NASA Contractor Report 189653, Volume II (Revised)

IN-16
150973
P-73

# Interplanetary Program To Optimize Simulated Trajectories (IPOST)

*Volume II - Analytic Manual*

P. E. Hong, P. D. Kent, D. W. Olson,
and C. A. Vallado

Martin Marietta Astronautics
Space Launch Systems Company
Denver, Colorado

Contract NAS1-18230
October 1992

NASA
National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

# TABLE OF CONTENTS

# 1.0 INTRODUCTION

The Interplanetary Program to Optimize Simulated Trajectories (IPOST) is intended to support many analysis phases, from early interplanetary feasibility studies through spacecraft development and operations. The IPOST output provides information for sizing and understanding mission impacts related to propulsion, guidance, communications, sensor/actuators, payload, and other dynamic and geometric environments.

Much of the overall architecture for IPOST has been derived from the Program to Optimize Simulated Trajectories (POST) (Reference 1-1). Indeed certain POST parameters and capabilities have been incorporated into IPOST to aid in POST-IPOST user compatibility. IPOST has extended trajectory capabilities to target planets and other celestial bodies with intermediate and velocity correction maneuvers. IPOST capabilities and limitations are summarized in Table 1-1.

| FEATURE | CAPABILITY |
|---|---|
| Optimization method | Explicit (Master/subproblems), Implicit (collocation) |
| Optimization algorithm | NPSOL |
| Optimization parameter* | ΔV magnitude, mass, time, . . . |
| Maximum controls | 25 (Master), 45 (subproblems), 1700 (collocation) |
| Control parameters* | Values of event criteria, ΔV, arrival conditions, thrust, . . . |
| Maximum targets | 25 (Master), 45 (subproblems), 1700 (collocation) |
| Target parameters* | Time, position, velocity, orbital conditions, . . . |
| Targeting method | NPSOL, Newton-Raphson, special Onestep |
| Sensitivity matrix | Finite differencing, analytic for special interplanetary targeting |
| Maximum events | 100 |
| Event criteria* | Time, distance, speed, closest approach, . . . |
| Event activities | Info, impulsive ΔV, launch, orbit insertion, mass jettison |
| Maximum maneuvers/ subproblems | 15 |
| Trajectory propagation | Conic, Onestep, Multiconic, Encke, Cowell, implicit |
| Planetary bodies | Sun, nine planets, Earth's moon, any user-defined bodies |
| Ephemeris | Analytic, precision (JPL) |
| Trajectory perturbations | Central body, perturbing bodies, radiation pressure, J2, aerodynamics, thrust |
| Input/Output frames | Ecliptic or planet equator, Mean 1950 or Mean 2000 |
| * User selectable | |

Table 1 - 1. IPOST Features/Capabilities

IPOST, along with members of its family, such as POST and IPREP, can analyze and support almost every activity associated with space exploration.

IPOST is event driven. That is, the user defines a sequence of events which are executed in the simulation process. The events can be triggered by different criteria, such as absolute or relative time, distance from a body, or propellant consumption. At the event times, various activities can be initiated or terminated, such as employing a different thrust steering law, changing trajectory propagators or propagation step size, performing an impulsive delta velocity maneuver or jettisoning a probe or stage.

The time period between two contiguous events is called a phase. Trajectory propagation takes place in each phase. Five types of propagators are available (listed in order of increasing accuracy and decreasing computational speed): Conic, Onestep, Multiconic, Encke, Cowell. Propagator selection depends upon user needs, such as simple fast simulations for parametric feasibility analysis, or precision detailed trajectories to support subsystem design.

IPOST can run a single trajectory simulation or it can run multiple simulations. For multiple simulations, one can run a parametric scan and/or an optimization mode. The search mode will vary one parameter, such as planetary arrival time, over a specified interval and increment size, and perform a simulation (or optimization) for each search parameter value.

The optimization mode will optimize a user cost/objective function, such as maximum mass that can be placed in a desired orbit, subject to user-specified constraints. The constraint variables, such as periapsis altitude or orbital inclination, are called dependent variables or target parameters. The parameters which are free to vary, such as maneuver delta velocity ($\Delta V$), are called independent variables or control parameters. As part of, or instead of, optimization, trajectory targeting can be performed. In this case, there is no cost function and the IPOST problem reduces to finding a set of control parameter values that meet specified target parameter conditions.

Generalized targeting and optimization uses the Stanford NPSOL algorithm. For certain types of problems, a trajectory decomposition method is available. There is a master optimization process which requires that the trajectory be divided into legs or sub-problems. Each subproblem is an optimization problem in itself, containing controls, constraints and an (optional) objective function. A special application of decomposition is the Interplanetary Targeting and Optimization Option (ITOO). This technique uses analytical partials generated during nominal trajectory propagation to determine minimum $\Delta V$ (or mass) trajectories, usually for gravity assist (swingby) missions.

In addition to the classic method of explicit optimization, there exists an option to perform implicit optimization using the collocation method. In this case, each phase is divided into independent segments which are allowed to vary subject to intersegment continuity and the equations of motion. Optimization using collocation is less sensitive to faulty initial guesses, but requires much greater CP time than explicit optimization to achieve the same level of accuracy.

IPOST input is via three namelists: $TOP, $TRAJ and $TAB. $TOP contains a description of the targeting and optimization problem. It must be input first. $TRAJ contains data that describes each mission event/phase. It must follow $TOP , and there must be one $TRAJ for each event. $TAB is used to input tabular data such as thrust vs. time or drag coefficient vs. mach number and angle of attack. Input and output units are metric.

# 2.0 COORDINATE SYSTEMS

There are many types of coordinate systems used in mission analysis applications. IPOST provides a number of systems to allow the user a maximum amount of analysis insight and flexibility.

## 2.1 INERTIAL ECLIPTIC SYSTEM

This system is used during heliocentric (sun-centered) interplanetary flight, although it is sometimes used as a fixed reference frame for the entire mission.

| | | |
|---|---|---|
| Center of system | - | the sun, planetary bodies, or planetary satellites |
| Primary plane | - | Earth ecliptic |
| X - axis | - | Earth vernal equinox direction |

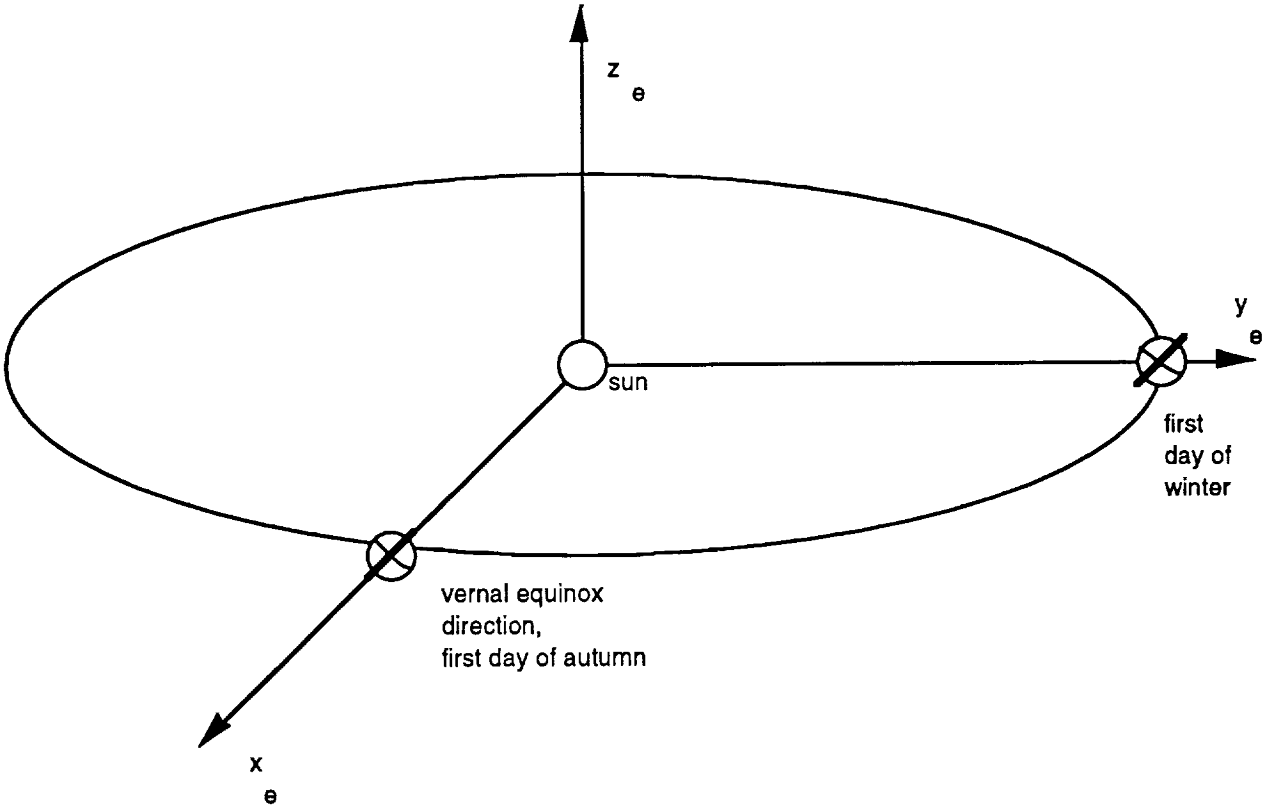


Figure 2 - 1. The inertial ecliptic coordinate system.
(The center can also be at any of the planetary bodies and satellites)

## 2.2 INERTIAL PLANETARY EQUATOR SYSTEM

This system is used during flight near a planetary body.

| | | |
|---|---|---|
| Center | - | planetary body |
| Primary Plane | - | planetary equator |
| X - axis | - | rotation of planetary vernal equinox direction through right ascension and declination angles of the planetary pole vector. |

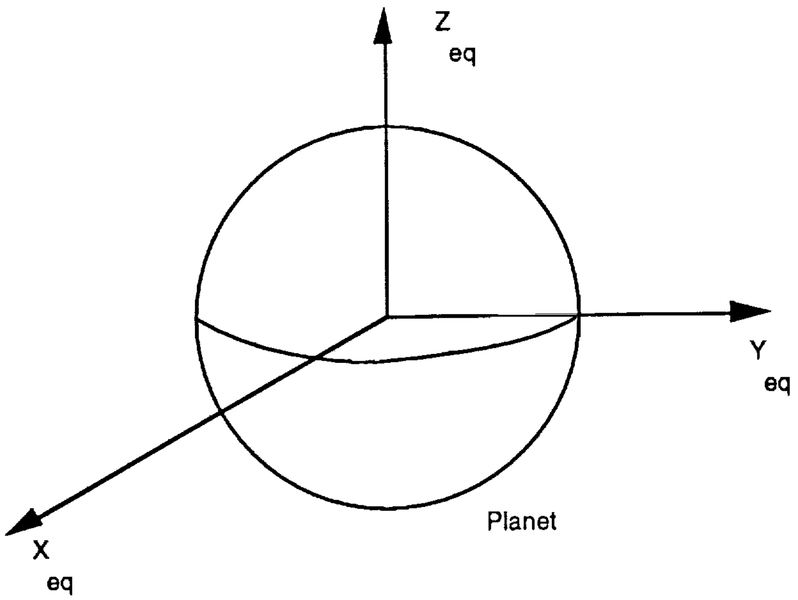
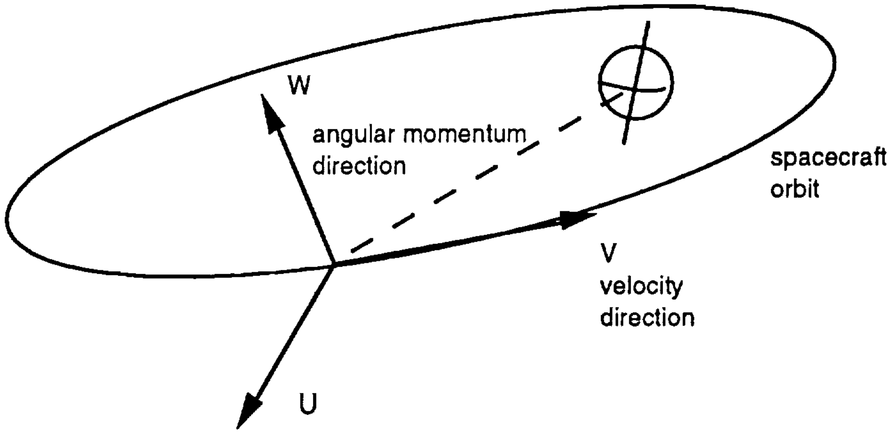


Figure 2 - 2. The planet equatorial coordinate system.

## 2.3 UVW SYSTEM

This system is used for vehicles whose longitudinal axis or whose thrust axis is along the velocity vector.

| | | |
|---|---|---|
| Center | - | S/C center of mass |
| Primary Plane | - | orbital plane of S/C |
| Primary axis (X) | - | cross product of S/C velocity U = V x W direction and S/C angular momentum direction |



$$V = \frac{\vec{v}}{|\vec{v}|}$$

$$W = \frac{\vec{r} \times \vec{v}}{|\vec{r} \times \vec{v}|}$$

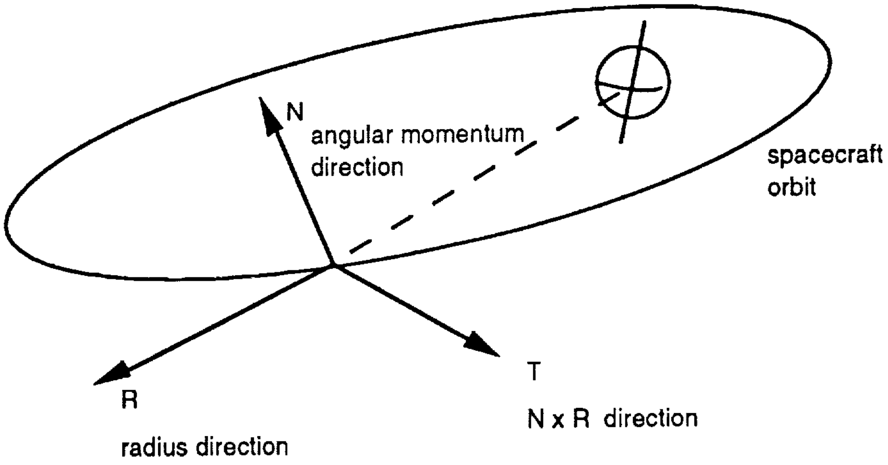
$$U = V \times W$$

Figure 2 - 3. The UVW Coordinate System.

## 2.4 RTN SYSTEM

This system is used for vehicles whose longitudinal axis is along the local vertical, such as gravity gradient stabilized s/c.

| | | |
|---|---|---|
| Center | - | S/C center of mass |
| Primary Plane | - | orbital plane of S/C |
| X - axis | - | radius vector direction |



$$\vec{R} = \frac{\vec{r}}{|\vec{r}|}$$

$$\vec{N} = \frac{\vec{r} \times \vec{v}}{|\vec{r} \times \vec{v}|}$$

$$\vec{T} = \vec{N} \times \vec{R}$$

Figure 2 - 4. The RTN Coordinate System.

## 2.5 THE BODY FRAME COORDINATE SYSTEM

The body frame is used in conjunction with other reference frames to orient the vehicle in celestial space, and to identify locations and orientations of vehicle components, such as the primary thrust vector and antenna boresights.

| | | |
|---|---|---|
| Center | - | S/C center of mass |
| $x_b$ axis | - | from center of mass through nose of S/C, |
| $z_b$ axis | - | from center of mass through bottom of S/C, orthogonal to x |
| $y_b$ axis | - | $\hat{x}_b$ x $\hat{z}_b$ |
| P | - | pitch |
| R | - | roll |
| Y | - | yaw |

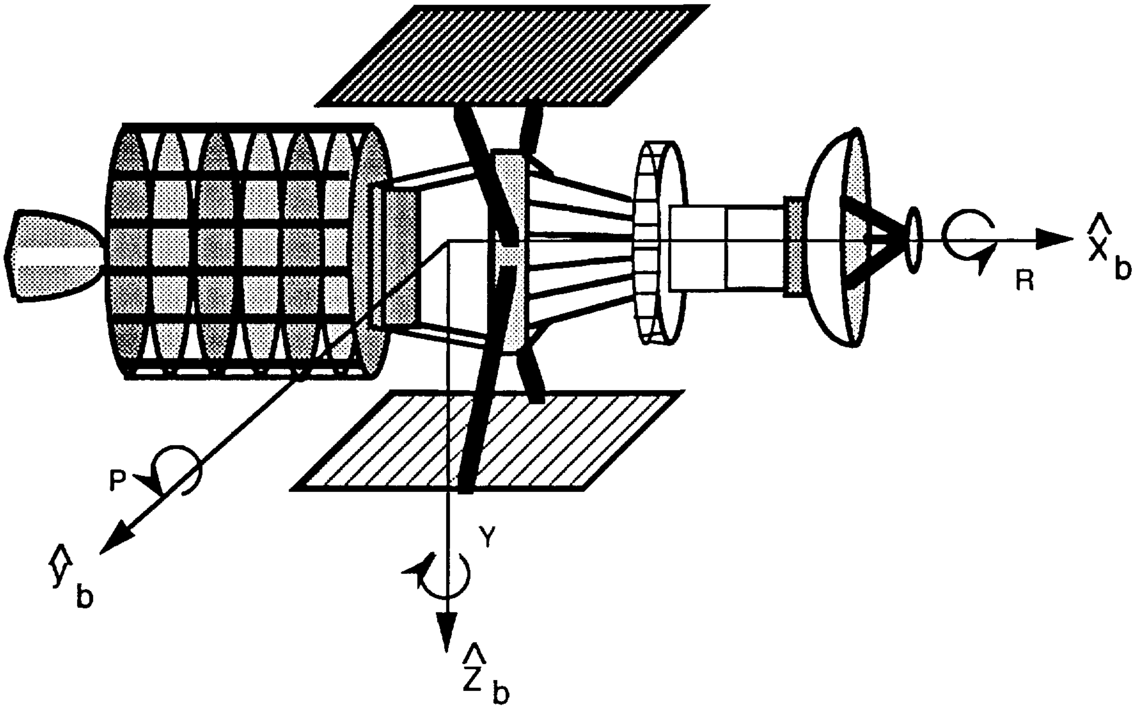
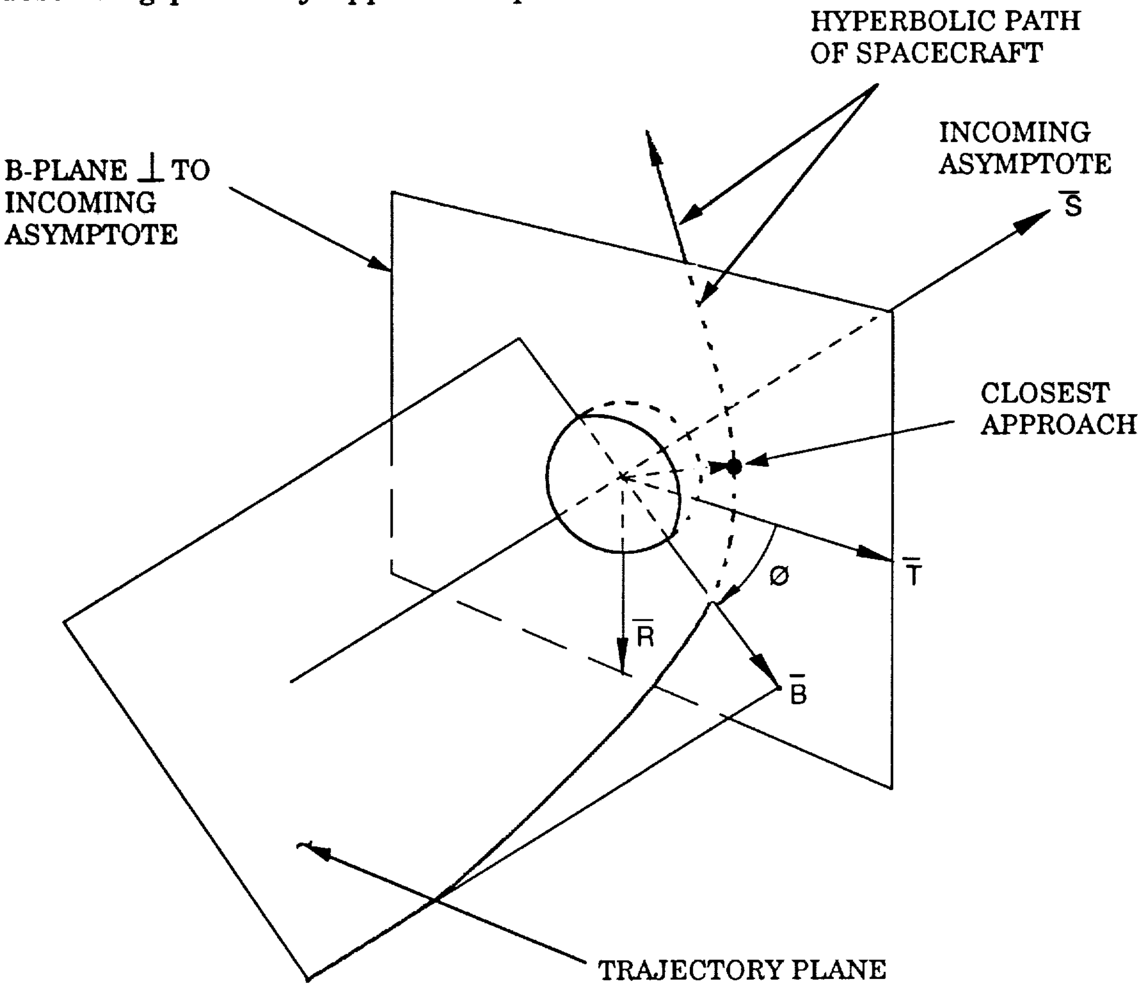


Figure 2 - 5. The body coordinate system.

## 2.6 THE B-PLANE

The B-plane coordinate frame is used for hyperbolic approach to, and departure from, a celestial body. It is often the most numerically stable system for describing planetary approach/departure conditions.



$\vec{B}$ = Impact Parameter (Vector from Planet Center to Aiming Point)

Ø = Orientation of $\vec{B}$ Relative to $\vec{T}$

$\vec{S}$ = Parallel to Incoming Asymptote

$\vec{T}$ = Parallel to Reference Plane (Ecliptic Unless Otherwise Specified)

$\vec{R} = \vec{S} \times \vec{T}$

Figure 2 - 6. B-Plane Coordinate System

## 2.7 CONE-CLOCK

The cone-clock system is used for determining the orientation of vehicle sensors and actuators. One application is to transform IPREP and QTOP thrust data into IPOST usable thrust acceleration profiles.

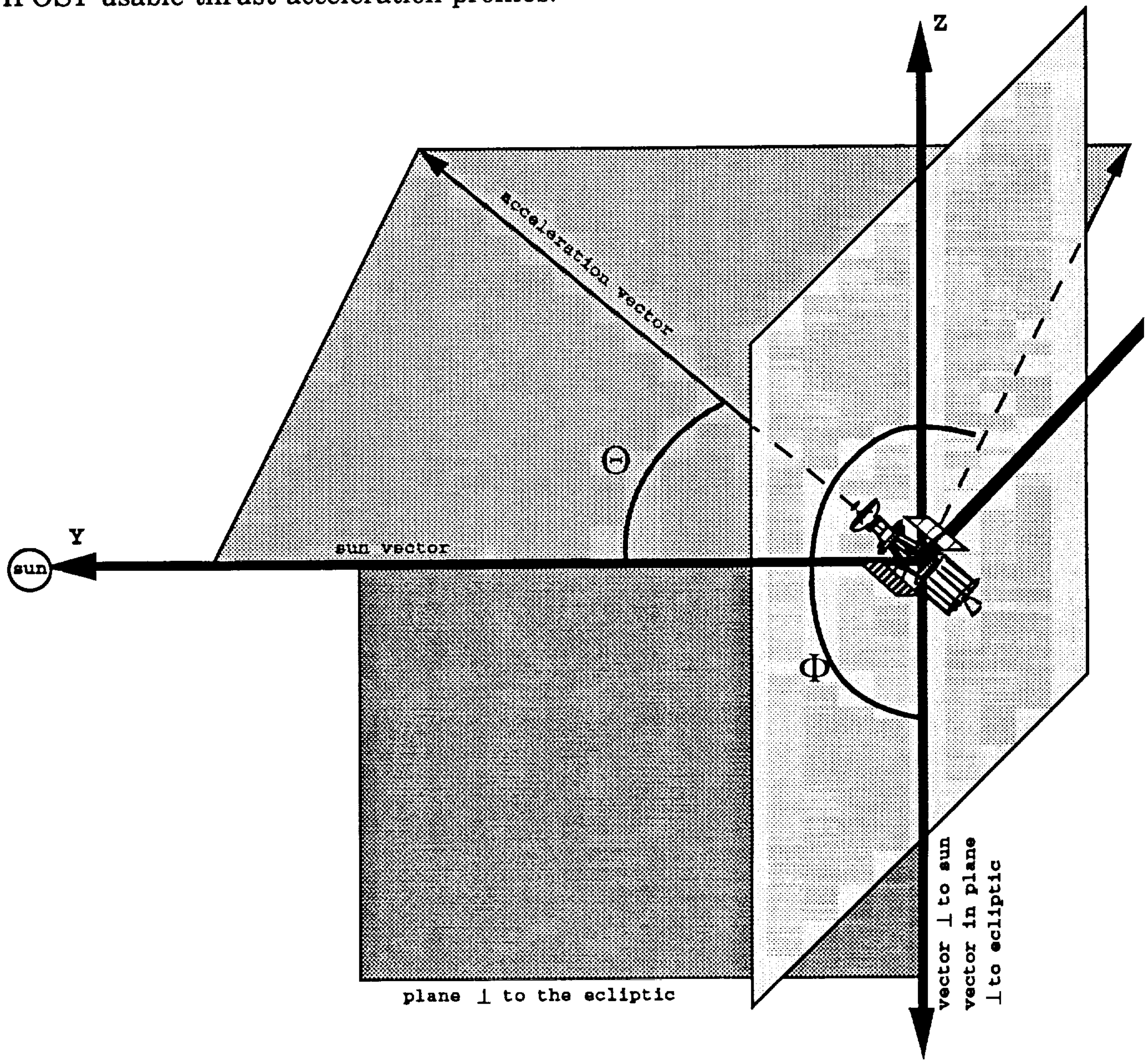


Figure 2 - 7. Cone/Clock Coordinate System

It can be seen from Figure 2 - 7 that to convert this vector to the heliocentric ecliptic (h/e) system, two rotations must be made. First, a rotation about xcc by the declination of the sun centered heliocentric ecliptic s/c vector, and then a rotation about the new z to line up the x and y axes. From this new vector in the h/e system, pitch and yaw can be calculated for the s/c orientation. This is input to IPOST. Since the cone clock system is not an inertial system, even if the cone and clock angles were fixed, the h/e acceleration vector would be continuously changing. Therefore, IPREP will calculate, for the IPOST user, cubic polynomial coefficients for yaw and pitch, as well as the acceleration magnitude, which can be input to IPOST as well. These polynomials are time dependent and can be used as controls in IPOST to "fine tune" trajectories due to modelling differences when higher orders of accuracy are desired.

The orientation of the body system relative to one of the other systems is given through rotations of the following Euler angles: roll, yaw, and pitch. These three angles are defined by time dependent quadratics where

$$\text{roll} = \phi = \phi_1 + \phi_2 t + \phi_3 t^2 \qquad ,[2\text{-}1]$$

$$\text{yaw} = \psi = \psi_1 + \psi_2 t + \psi_3 t^2 \qquad ,[2\text{-}2]$$

$$\text{pitch} = \theta = \theta_1 + \theta_2 t + \theta_3 t^2 \qquad ,[2\text{-}3]$$

The rotation matrices for body system to system for which roll, yaw, and pitch are defined as:

$$\begin{bmatrix} \cos\theta \cos\psi & \cos\theta \cos\phi \sin\psi + \sin\theta \sin\phi & \cos\theta \sin\phi \sin\psi + \sin\theta \cos\phi \\ -\sin\psi & \cos\phi \cos\psi & \cos\phi \sin\psi \\ \sin\theta \cos\psi & \sin\theta \cos\phi \sin\psi - \cos\theta \sin\phi & \sin\theta \sin\phi \sin\psi + \cos\theta \cos\phi \end{bmatrix}$$

UVW system to planet equatorial system:

$$\begin{bmatrix} U_x & V_x & W_x \\ U_y & V_y & W_y \\ U_z & V_z & W_z \end{bmatrix}$$

RTN system to planet equatorial system:

$$\begin{bmatrix} R_x & T_x & N_x \\ R_y & T_y & N_y \\ R_z & T_z & N_z \end{bmatrix}$$

Since each of these matrices consists of orthogonal rotations, the inverse rotation is merely the transpose of the matrix.

# 3.0 CONIC PROPAGATORS

IPOST uses two classical propagators for two-body motion: GOODYR and LAMBRT. Both propagators can handle elliptical (including multiple revolutions) and hyperbolic orbits.

GOODYR computes a final cartesian state given an initial state and a time of flight. The algorithm is based on Goodyear's method (Reference 3-1). The two-body equations of motion are solved using a generalized eccentric anomaly and generalized Kepler's equation. The final state is computed using f and g series. The state transition matrix, and its inverse, are similarly computed.

LAMBRT computes the initial and final velocities given the initial and final positions and a time of flight. This classical Lambert problem is solved using a method described in Reference 3-2. A generalized time of flight is iterated upon using an independent parameter which also depends on known orbit quantities.

# 4.0 ONESTEP (1STEP) PROPAGATOR

ONESTEP is a special adaptation of a generalized Multiconic method. It is used in JPL's PLATO, and described in References 4-1 and 4-2. The 1STEP computational sequence is as follows (see also Figure 4-1).

(1) Start with an initial state at $t_i$ and a desired propagation final time $t_f$. There are other specifications needed, such as identifying the primary and secondary bodies and the sphere of influence time ($\Delta t$).

(2) Estimate $t_p$, the closest approach time at the secondary body, by Lagrangian interpolation using 3 radii (at initial, final, and midpoint times) based on primary body conic motion.

(3) Compute the new corrected closest approach time $t_p'$ by conic propagation.

(4) If $t_p$ and $t_p'$ are within an acceptable tolerance, proceed to Step 5, otherwise, replace $t_p$ with $t_p'$ and go back to Step 2.

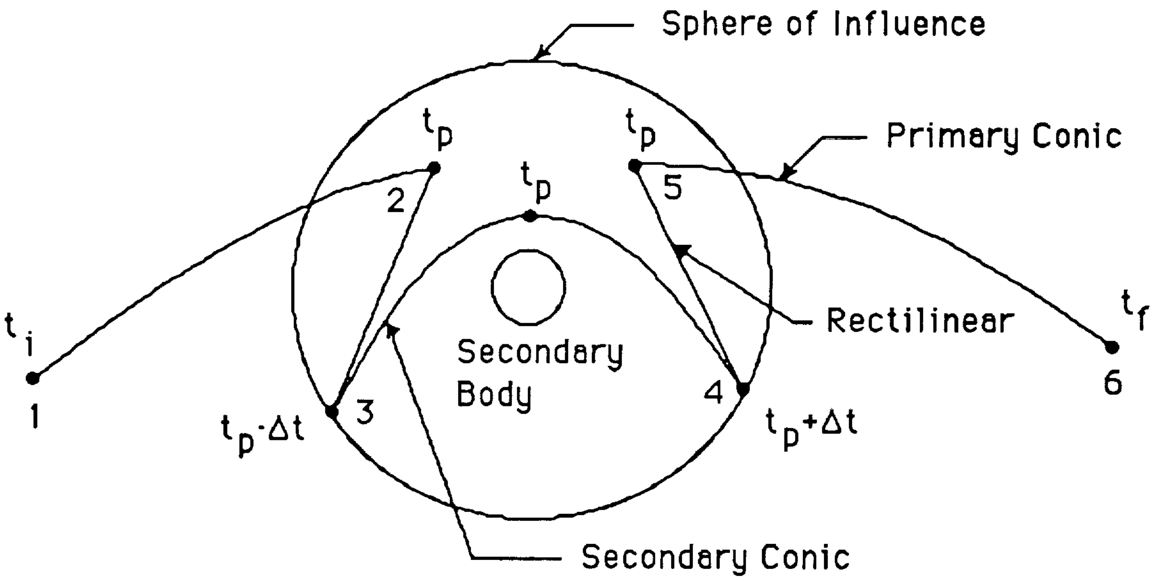


Figure 4-1. ONESTEP flyby process

(5) Propagate the initial state $t_i$ to $t_p$ using a two body conic (IPOST uses GOODYR) centered about the primary body. This state is called the (incoming) pseudostate. The propagation is from point 1 to point 2 in figure A-1.

(6) Transform the pseudostate to the secondary body frame.

(7) Propagate the state from $t_p$ to $t_p - \Delta t$ (backward in time) assuming the velocity at $t_p$ is constant. This is the rectilinear segment from point 2 to point 3.

(8) Propagate the state at $t_p - \Delta t$ to $t_p + \Delta t$ using a two body conic (GOODYR) centered about the secondary body.

(9) Propagate the state from $t_p + \Delta t$ backwards to $t_p$ assuming the velocity at $t_p + \Delta t$ is constant.

(10) Transform this (outgoing pseudostate) state to the primary body frame.t

(11) Propagate the pseudostate from $t_p$ to $t_f$ using a two body conic (GOODYR) centered at the primary body. This is the final output state. The final propagation is illustrated as the segment from point 5 to point 6 in Figure 4-1.

There are also tests and modifications performed if the final time is before $t_p$, or if the initial time is after $t_p$, or if the trajectory is totally outside of the secondary body sphere of influence.

The targeting and optimization process usually requires a state transition matrix. When 1STEP is used as a propagator, the state transition matrix can be computed analytically. This is done by chaining a sequence of separate transition matrices, all of which are computed analytically.

$$\Phi_{fi} = \partial X_f / \partial X_i = \Phi_{65} \Phi_{54} \Phi_{43} \Phi_{32} \Phi_{21} \quad , \quad [4\text{-}1]$$

where $\Phi_{65}$, $\Phi_{43}$, $\Phi_{21}$ are conic transition matrices generated as part of the GOODYR conic propagation in Steps 11, 8, and 3, respectively. The constant velocity transition matrices are

$$\Phi_{54} = {}^{-}\Phi_{32} = \begin{pmatrix} I & I\Delta t \\ 0 & I \end{pmatrix} \quad , \quad [4\text{-}2]$$

# 5.0 MULTICONIC

The Multiconic propagator is similar to ONESTEP (Section 4) except that it takes multiple steps to traverse a given time interval. This is done to avoid the restrictive assumptions of ONESTEP, particularly having to always fly to closest approach of the secondary body, and to accommodate additional perturbations. The algorithm solves the three body problem following References 5-1, 5-2, 5-3. It is non-directed, meaning that it does not presume a direction toward or away from either the primary or secondary bodies. It is presumed that the secondary body is in orbit about the primary body.

The algorithm is straightforward. We are interested in propagating a S/C state $[\, R_I, \dot{R}_I \,]$ with respect to a primary body at time $t_I$ to another time, $t_F = t_I + h$ in the presence of a secondary body. The solution is

$$R_F = R_{IPF} + r_{ISF} - r_I - h\dot{r}_I + \mu \; (\rho_F - \rho_I - h\,\dot{\rho}_I) + \frac{1}{2}\; a_p h^2 \qquad , [5\text{-}1]$$

$$\dot{R}_F = \dot{R}_{IPF} + \dot{r}_{ISF} - \dot{r}_I + \mu \left(\dot{\rho}_F - \dot{\rho}_I\right) + h a_p \qquad , [5\text{-}2]$$

where $R_{IPF}$ = position vector of S/C with respect to primary body mapped with a 2 body conic from $t_I$ to $t_F$.

$r_{ISF}$ = position vector of S/C with respect to secondary body mapped with a 2 body conic from $t_I$ to $t_F$.

$r_I$ = position vector of S/C with respect to secondary body at $t_I$.

$\rho_I$ = position vector of the secondary body with respect to the primary body at $t_I$.

$\mu$ = $\mu_S/(\mu_P+\mu_S)$

h = the delta time, $t_F - t_I$.

$a_p$ = the average perturbing acceleration over $\Delta t$.

The perturbing acceleration is the sum of the radiation pressure and the gravitational perturbations from bodies other than the primary and secondary bodies and other forces. Section 8 describes the perturbing acceleration models.

There is an optional variable stepsize algorithm,

$$h = \left( \frac{S_0 \, R_P^3 \, R_{SMIN}^2}{\mu_P \quad \mu_{SMIN}} \right)^{1/3} \qquad , [\ 5\text{-}3]$$

where $S_0$ is a user input scale factor,

$R_P$ = distance of S/C from primary body,

$R_{SMIN}$ = distance of S/C from closest secondary or perturbing body,

$\mu_{SMIN}$ = gravitational constant of closest secondary or perturbing body

The state transition matrix, if needed, is given by

$$\Phi\,(t_F, t_I) = \Phi_{2B}^{P}\,(t_F, t_I) + \Phi_{2B}^{S}\,(t_F, t_I) - \begin{bmatrix} I & hI \\ O & I \end{bmatrix} \qquad , [5\text{-}4]$$

Where $\Phi_{2B}^{P}$ is the two-body transition matrix from $t_I$ to $t_F$ about the primary body,

$\Phi_{2B}^{S}$ is the two-body transition matrix from $t_I$ to $t_F$ about the secondary body.

# 6.0 ENCKE PROPAGATOR

For more precision trajectory propagation, the Encke method has been adapted as described in Reference 6-1. Instead of integrating the sum of all accelerations, as in the Cowell method, this method integrates the difference between the true orbit and an unperturbed conic orbit. Therefore, only the perturbations are integrated. Since the perturbations change much more slowly than the actual state, a larger step size can be utilized, making the method 3 to 10 times faster than the Cowell method, with a reduction in roundoff error. The reference conic orbit is the orbit which would result if all perturbing accelerations were removed at a particular time.

The reference orbit is used for calculating the difference in total accelerations, until this difference becomes too large. When this occurs, a process called rectification occurs so that integration may continue without loss in accuracy. This means that a new epoch is chosen, and a new reference orbit is calculated which coincides with the true orbital path at that time.

The objective of the Encke method is to find an analytic expression for the difference in the acceleration vectors of the true and the reference orbits. First, let $\vec{\rho}$ denote the state vector of the S/C on the true orbit, and let $\vec{r}$ denote the state of the reference orbit at the same time. The equations of motion for these two states are then

$$\frac{d^2\vec{r}}{dt^2} + \frac{\mu}{r^3}\vec{r} = \vec{a}_p \quad , \qquad [6\text{-}1]$$

and

$$\frac{d^2\vec{\rho}}{dt^2} + \frac{\mu}{\rho^3}\vec{\rho} = 0 \quad , \qquad [6\text{-}2]$$

where $\vec{a}_p$ is the vector sum of all perturbing accelerations. The difference vector of the true to the reference orbit is then

$$\delta\vec{r} = \vec{r} - \vec{\rho} \quad , \qquad [6\text{-}3]$$

and the second derivative with respect to time is given as

$$\delta \vec{r}'' = \vec{r}'' - \vec{\rho}'' \quad . \qquad [6\text{-}4]$$

For initial conditions, recall that at the epoch, $t_0$ , the state of the true and reference orbits are coincident, thus

$$\vec{r}\,(t_0) = \vec{\rho}(t_0) \text{ and } \vec{r}'\,(t_0) = \vec{\rho}'\,(t_0) \quad . \qquad [6\text{-}5]$$

By substituting equations [6-1] and [6-2] into equation [6-4], we arrive at the analytic expression

$$\delta \vec{r}'' = \vec{a}_p + \left[ \frac{\mu}{\rho^3} \vec{\rho} - \frac{\mu}{r^3} \vec{r} \right] \qquad [6\text{-}6]$$

which can be rewritten as,

$$\delta \vec{r}'' = \vec{a}_p + \frac{\mu}{\rho^3} \left[ \left( 1 - \frac{\rho^3}{r^3} \right) \vec{r} - \delta \vec{r} \right] \qquad [6\text{-}7]$$

The above equation presents some numerical difficulties. The reason for using this method instead of the Cowell method was to obtain more accuracy, through less roundoff error. The expression $( 1 - \rho^3 / r^3)$, with its very nearly equal terms, is difficult to evaluate numerically, requiring many more digits of accuracy.

In reference 6-1 the problem was treated as follows.

Let

$$-f(q) = 1 - \frac{\rho^3}{r^3} = 1 - (1 + q)^{3/2} \qquad [6\text{-}8]$$

where

$$q = \frac{(\delta \vec{r} - 2 \vec{r}\ ) \bullet \delta \vec{r}}{r^2} \qquad [6\text{-}9]$$

and

$$f(q) = q \frac{3 + 3q + q^2}{1 + (1 + q)^{3/2}} \qquad [6\text{-}10]$$

Then, equation (6) becomes

$$\delta \vec{r}'' = - \frac{\mu}{\rho^3} \left[ f(q) \vec{r} + \delta \vec{r} \right] + \vec{a}_p \qquad [6\text{-}11]$$

which removes the numerical problem, and is the equation integrated in the Encke method.

The Encke method reduces the number of integration steps needed on a given interval since $\delta \vec{r}$ changes much more slowly than $\vec{r}$. If the acceleration due to the perturbations approaches that of the central body term, or if $\delta \vec{r} / \vec{r}$ does not remain small, the advantages of the method diminish rapidly. As the perturbations become quite large, then it is possible the reference parameters should be changed since the perturbations are becoming primary, or the Cowell method should be employed. In the case that $\delta \vec{r} / \vec{r}$ has become large, the reference orbit more than likely needs to be rectified.

## 6.1 ENCKE STATE TRANSITION MATRIX FORMULATION

As an option in the Encke method, the state transition matrix, $\Phi$ , can also be integrated along with the state.

From [6-1] we have

$$\Phi = \frac{\partial \vec{x}(t)}{\partial \vec{x}(t_0)} \qquad [6\text{-}12]$$

where $\vec{x}(t)$ is the state at time t, and $\vec{x}(t_0)$ is the state at the initial time, $t_0$. When $t = t_0$, the matrix is a 6x6 identity matrix. The derivative of $\Phi$ is given as

$$\Phi' = F\ \Phi \qquad [6\text{-}13]$$

where

$$F = \frac{\partial \vec{x}'(t)}{\partial \vec{x}(t)} \quad . \qquad [6\text{-}14]$$

If $\vec{r}$ is the radius vector, $\vec{r}'$ is the velocity vector, and $\vec{r}''$ is the acceleration vector, then

$$F = \begin{bmatrix} \frac{\partial \vec{r}''}{\partial \vec{r}} & \frac{\partial \vec{r}'}{\partial \vec{r}'} \\ \frac{\partial \vec{r}''}{\partial \vec{r}} & \frac{\partial \vec{r}''}{\partial \vec{r}'} \end{bmatrix} \quad . \qquad [6\text{-}15]$$

Therefore we have the 6x6 matrix

$$F = \left[\begin{array}{c|c} \phi & I \\ & \\ B & C \end{array}\right] \qquad [6\text{-}16]$$

where

$\phi$ = 3x3 null matrix,
I = 3x3 identity matrix,

$$B = \frac{\partial \vec{r}\,''}{\partial \vec{r}}, \text{ and} \qquad [6\text{-}17]$$

$$C = \frac{\partial \vec{r}\,''}{\partial \vec{r}\,'} \quad . \qquad [6\text{-}18]$$

For each set of perturbation equations, the matrices B and C are calculated, and used to create the matrix F. Once F is calculated for all perturbations, $\Phi'$ can then be integrated along with the state.

Perturbations which are accounted for in the transition matrix computation include perturbing bodies, $J_2$, and atmospheric drag.

## 6.2 ENCKE VARIABLE TIME STEP FORMULATION

The Encke method also has the option of a variable time step which is dependent on the size of the perturbations. Reference 6-2 suggests that the time step should be dependent on the B partition of the F matrix derived in the preceding section. The time step for input to the numerical integrator (RUNGE4) is defined as

$$h = (\,|\,B_1\,|^2 + |\,B_2\,|^2 + \,|\,B_3\,\,|^2\,)^{.25} * \mathrm{TOL} / \mathrm{SPDY} \qquad [6\text{-}19]$$

where

$B_1$ is the first vector column of B,
$B_2$ is the second vector column of B,
$B_3$ is the third vector column of B,
TOL is the error tolerance level, and
SPDY is the variable for seconds per day.

# 7.0 COWELL PROPAGATOR

The Cowell method is the most straightforward of all the numerical trajectory propagation methods which include perturbations. The application of the method is simply to write the equations of motion of the studied object, including all perturbations, and then to integrate them step-by-step numerically. The equations of motion for a spacecraft would be

$$\vec{r}'' + (\mu / r^3)\,\vec{r} = \vec{a}_p \qquad [7\text{-}1]$$

where $a_p$ is the vector sum of all the perturbative accelerations. This equation is broken down, for numerical integration, into a set of first order vector component equations.

$$x' = \vec{v}_x \qquad \vec{v}_x{}' = \vec{a}_{px} - (\mu / r^3)\,x \qquad [7\text{-}2]$$

$$y' = \vec{v}_y \qquad \vec{v}_y{}' = \vec{a}_{py} - (\mu / r^3)\,y \qquad [7\text{-}3]$$

$$z' = \vec{v}_z \qquad \vec{v}_z{}' = \vec{a}_{pz} - (\mu / r^3)\,z \qquad [7\text{-}4]$$

Advantages of the Cowell method come from its simplicity of formulation and implementation. The advantages are equally weighted, though, by a set of disadvantages. When the acceleration forces are large (such as motion near a large attracting body), decreasingly smaller time steps must be taken which severely affect the speed at which the method operates. Also, when such small steps are taken, there can be a large accumulation of roundoff error. Roundoff error will also begin to accumulate in interplanetary flight if the step size taken is quite small. This gives support to the use of the Encke and Multiconic methods for interplanetary flight and for low perturbation trajectories. When the perturbations become quite large though, the Cowell method must be utilized.

# 8.0 PERTURBING ACCELERATIONS

Perturbing accelerations include radiation pressure, low and high thrust propulsion, aerodynamics, gravitational nonsphericity of the central body and perturbations resulting from multiple attracting bodies.

## 8.1 RADIATION PRESSURE

Acceleration due to solar radiation pressure is modeled by

$$\vec{a} = \frac{S_0 C_R}{m\, r^2} \left(\vec{A}_{SC} \cdot \vec{r}\right) \vec{r}, \qquad [8\text{-}1]$$

where

$S_0$ = Solar flux constant,
$C_R$ = Coefficient of reflectivity,
$m$ = Mass of S/C,
$\vec{r}$ = Heliocentric radius unit vector,
$\vec{A}_{SC}$ = Effective area used to calculate radiation press
(orthogonal to body axes, transformed into the i
frame).

## 8.2 PROPULSION

There are three types of propulsion systems that are modeled: low thrust, generalized, and blowdown. Low thrust propulsion accelerates ions that are supplied through electrical power systems, nuclear or solar. Generalized systems are traditional high thrust propulsion systems used in a variety of rocket applications, from launch vehicles to S/C attitude control. Blowdown systems are a relatively economical S/C propulsion system.

All three propulsion systems provide both acceleration and propellant mass flow information to the propagators. The mass flow is integrated simultaneously with the translational equations of motion, using the mass flow equations that are applicable for each propulsion system.

The thrust direction is assumed to be fixed with respect to the body. The thrust direction is given as a unit vector in the body frame.

Acceleration due to low thrust electric propulsion is modeled as

$$\vec{a} = \frac{2\, u\, \eta\, P}{m\ g\, I_{sp}} \vec{u}, \qquad [8\text{-}2]$$

and the mass flow rate is given as

$$\dot{m} = \frac{m\ |\ \vec{a}\ |}{g\ I_{sp}}$$

where

$u$ = Throttle level (normalized),
$\eta$ = Thruster efficiency,
$g$ = 9.806 m/s,

$I_{sp}$ = Thruster specific impulse,
$\vec{u}$ = Thrust direction,
$P$ = Electric power available to thruster.

The available power for the thruster system is

$$P = P_0\left(1 + \frac{C1}{r^2} + \frac{C2}{r^3} + \frac{C3}{r^4}\right)e^{-P_L(t - to)} - P_{HK}, \qquad [8\text{-}3]$$

where

$P_0$ = Initial power, watts,
$C_1, C_2, C_3$ = Solar array efficiency factors (=0 for NEP),
$P_L$ = Reactor decay factor (=0 for SEP),
$t_0$ = Initial time,
$P_{HK}$ = Housekeeping power for space vehicle, watts.

Two finite thrust models will be employed. The first is a generalized system which can be a regulated pressure system in which a separate gas tank provides constant pressure to the propellant tank, or a solid rocket. The generalized finite thrust is modeled as

$$T = u\,T_{vac} - A_E P(h) \;,$$

$$\dot{m} = \frac{u\,T_{vac}}{g\,I_{sp}} \;,$$

$$\vec{a} = \frac{T}{m}\,\vec{u} \;, \qquad [8\text{-}4]$$

where

$I_{sp}$ = Thruster specific impulse,

$\dot{m}$ = Mass flowrate of propellant to thruster(s),
$u$ = Throttle level (normalized),
$\vec{u}$ = Thrust direction,
$m$ = Mass of the S/C,
$T_{vac}$ = Vacuum thrust of the engine(s), Newtons,
$A_E$ = Exit area, $m^2$,
$P(h)$ = Atmospheric pressure at altitude, h.

A blowdown system will also be modeled. A blowdown system uses one pressurized tank partially filled with propellant. The system is modeled as

$$\dot{m} = u \;\; \dot{m}_{max},$$

$$\vec{a} = \frac{T_0\, u\, \vec{u}}{m\left(1+\frac{u\,\dot{m}\,(t - t_0)}{\rho_f V_0\, U_l}\right)^{\gamma}}, \qquad [8\text{-}5]$$

$$T = m \;|\; \vec{a} \;|\,,$$

$$I_{sp} = T\,/\,(\dot{m}\; g)\,,$$

where

$T_0$ = Initial thrust level, Newtons,

$\dot{m}$ = Mass flowrate of propellant to thruster(s),

$\dot{m}_{max}$ = Maximum mass flowrate,

$u$ = Throttle level (normalized),

$\vec{u}$ = Thrust direction,

$m$ = Mass of the S/C,

$\rho f$ = Propellant density,

$t_o$ = Initial time,

$V_0$ = Initial tank volume,

$U_l$ = Ullage ratio,

$\gamma$ = Ratio of specific heats.

## 8.3 GRAVITATIONAL PERTURBATIONS

Disturbing body accelerations are modeled in the Multiconic, Cowell and Encke propagators as

$$\vec{a} = \sum_{i=1}^{n} \frac{\mu_i\left(\vec{r}_i - \vec{R}\right)}{|\;\vec{r}_i - \vec{R}\;|^3} \quad , \qquad [8\text{-}6]$$

where

$\mu_i$ = Gravitational constant of the ith body,

$\vec{r}_i$ = Position vector of the ith body (from ephemeris),

$\vec{R}$ = Position vector of the space vehicle,

$n$ = Number of perturbing bodies.

Gravitational potential for nonspherical J harmonics is modeled as

$$\phi = -\frac{\mu}{r}\left(\frac{J_2}{2}\right)\left(\frac{r_e}{r}\right)^2\left(3\sin^2 L - 1\right),$$

where $\mu$ = gravitational constant for the central body,

$L$ = planetocentric latitude,

$r_e$ = equatorial radius.

Therefore, gravitational accelerations due to $J_2$ in the planet centered frame are given as

$$x'' = \frac{\delta\phi}{\delta x} = -\frac{\mu x r_e^2}{r^5}\left(\frac{3}{2}\right) J_2 5 \left(\frac{1-5z^2}{r^2}\right) , \qquad [8\text{-}7]$$

$$y'' = \frac{\delta\phi}{\delta x} = \frac{y}{x}\ x'' , \qquad [8\text{-}8]$$

$$z'' = \frac{\delta\phi}{\delta z} = -\frac{\mu z r_e^2}{r^5}\left(\frac{3}{2}\right) J_2 \left(3 - 5\frac{z^2}{r^2}\right) , \qquad [8\text{-}9]$$

## 8.4 AERODYNAMICS

Aerodynamic accelerations are composed of drag and lift, with no side force. The drag and lift coefficients can range from constants to bivariant functions of mach number and angle of attack.

The atmospheric model used in aerodynamic calculations can be either a simple exponential density or pressure/temperature versus altitude. If the exponential atmosphere model is selected, certain aerodynamic effects will not be included, such as thrust back pressure and mach number variations.

Acceleration due to aerodynamic drag is modeled as

$$\vec{a} = -\frac{1}{2} C_D \frac{A}{M} \rho\, v_a \vec{r}\,'_a , \qquad [8\text{-}10]$$

where

$C_D$ = drag coefficient,

$A$ = cross sectional area of S/C perpendicular to the direction of motion

$m$ = S/C mass,

$\rho$ = atmospheric density at the altitude of the S/C,

$v_a$ = $|\ \vec{r}\,'_a\ |$ = speed of S/C relative to the rotating atmosphere,

$$\vec{r}\,'_a = \begin{pmatrix} x' + \theta' y \\ y' - \theta' x \\ z' \end{pmatrix} \quad ,$$

where

$$\vec{r}\,' = \begin{pmatrix} x' \\ y' \\ z' \end{pmatrix} = \text{inertial velocity in planet}$$

equatorial system,

and

$\theta'$ = rotation rate of the planet.

Acceleration due to aerodynamic lift is modeled as

$$\vec{a} = \frac{1}{2} C_L \frac{A}{m} \rho V_a^2 \hat{n}' \quad , \qquad [8\text{-}11]$$

where

$C_L$ = lift coefficient,
$\hat{n}'$ = unit vector in the lift direction.

As mentioned earlier, two types of atmospheric models are available for calculation of the density at a given altitude. The first is a simple exponential model where the density is

$$\rho = \rho_o e^{(-1/a_h)(h - h_o)} ,$$

where

$\rho_o$ = base density,
$a_h$ = scale height,
$h$ = current altitude,
$h_o$ = base altitude.

Currently stored in $\rho_o$, $a_h$ , and $h_o$ are the values for the Earth's atmosphere.

Also available is a more complex tabular atmospheric model which calculates, for a specific altitude, atmospheric density, atmospheric pressure, molecular weight, mach number, and molecular temperature. Stored tabular values are altitude, molecular weight, molecular temperature, and pressure. These values are interpolated to give the correct output values for density, pressure, weight, mach number and temperature. As with the exponential atmosphere, the stored values currently available are for the Earth.

# 9.0 ANALYTIC EPHEMERIS

The analytic ephemeris in IPOST is taken from MAPSEP (Reference 9-1). Nine planets with respect to the sun are represented. Each planet is characterized by six conic elements $(a,e,i,\omega,\Omega,M)$ in a heliocentric mean ecliptic 1950.0 frame of reference. Each conic element is described by a third order polynomial in time. For example, Mercury's inclination in radians is

$$i = .01222... + (-1.041...E\text{-}4)*T + (1.745...E\text{-}8)*T^2 + (0)*T^3$$

where time (T) is measured in Julian centuries past 2000.0 .

When the ephemeris routine (EPHEM) is called to compute the planet state (position and velocity), the conic elements are evaluated via their time polynomials. The conic elements are then transformed to cartesian space, and the state vector is output.

Analytic planetary satellite ephemeris is treated similarly. For example, the Earth's moon is described by six conic elements with respect to Earth, each of which has four polynomial coefficients describing the respective element in time. At any specified time, the lunar cartesian state is computed, and added to the Earth cartesian state to provide the moon's state with respect to the primary body.

# 10.0 PRECISION EPHEMERIS

Extremely accurate planetary and lunar positions and velocities are derived by evaluating a file of Chebyshev polynomials calculated/tailored for a given mission application. This is accomplished using multiple-day-arc Chebyshev polynomials whose coefficients are derived by a pre-processor using ephemeris data supplied by the Jet Propulsion Laboratory (JPL), Pasadena, California. These data files contain positions and velocities referred to the rectangular equatorial coordinate system of the mean equator and equinox of 2000 for the major solar bodies: Mercury, Venus, Earth- Lunar barycenter, Mars, Jupiter, Saturn, Uranus, Neptune, Pluto, and Earth's moon plus the nutation rates in longitude and obliquity. This data was generated at JPL by weighted, least-squares estimation of appropriate orbital models using source positions obtained on the basis of current planetary theories.

Using the method of Reference 10-1, a solar/lunar/planetary (S/L/P) file of Chebyshev polynomial coefficients are generated for the time period(s) of interest for each mission application. For trades performed for given mission(s), these files would be saved for later access to avoid recomputing the coefficients. In this manner, Chebyshev polynomial representations will retain the accuracy of the original JPL data while increasing computing efficiency by eliminating the need to interpolate on the JPL ephemeris data tapes. The Chebyshev file accessed during computations contains polynomial coefficients for each component of position and velocity and for each element of the matrix that transforms from the mean equator and equinox of date to the true of date coordinate system as required for IPOST calculations. Also included in the file would be the coefficients for the equation of equinoxes, $\Delta H$, used to correct mean Greenwich sidereal time.

In conjunction with the ephemerides pre-processor, a file-read utility is required to access the Chebyshev polynomials and to calculate the ephemerides for IPOST or to compute them in the coordinate system(s) of choice for output.

# 11.0 NPSOL

NPSOL is a sophisticated optimization package developed by Systems Optimization Laboratory of Stanford University (References 11-1 and 11-2). It minimizes a smooth cost function $F(\vec{y},\vec{u})$ where $\vec{u}$ are the control (independent) parameters subject to upper and lower bound constraints on both the independent variables and on dependent functions, $\vec{y}(\vec{u})$. These $\vec{y}$ can be linear and/or nonlinear functions, some of which may be target (dependent) variables.

NPSOL is a stand alone package which interfaces with IPOST by accepting current trajectory values (the cost function F, the cost gradient $\vec{g} = \partial F/\partial \vec{u}$, the cost Hessian $H = \partial^2 F/\partial \vec{u}^2$, controls $\vec{u}$, constraints $\vec{y}$, and the sensitivities $C = \partial \vec{y}/\partial \vec{u}$) and returning the next iteration control parameters $\vec{u}$ which minimize the cost function $F(\vec{u})$. This process is iterated to convergence, eventually outputing the optimum $\vec{u}$ and the values for F and $\vec{y}$.

The NPSOL algorithm (Reference 8) uses major and minor iterations to solve the targeting and optimization problem. In the major iterations NPSOL seeks a significant decrease in the merit function along a direction of search $\vec{p}$. NPSOL defines the merit function as:

$$F(\vec{u}) = \sum_i [\, li * (ci\ (\vec{u}) - si)] + \frac{1}{2}\sum_i\ [ri * \ (ci\ (\vec{u}) - si)^2]$$

where $F(\vec{u})$ is the objective function
li is the vector of lagrange multipliers
ci is the i-th constraint function
si is the set of slack variables used to handle inactive constraints and
ri is the penalty vector for constraint violations.

Therefore, during NPSOL operation, the problem becomes better targeted and more optimized in a simultaneous fashion.

In the minor iteration NPSOL seeks the search direction $\vec{P}$ for the major iteration by minimizing the quadratic programming problem:

$$Q = \vec{g}^T \vec{p} + \frac{1}{2} \vec{p}^T H \vec{p},$$

subject to a set of constraints, where $\vec{g}$ is the gradient of $F(\vec{u})$, and H is the quasi-Newton approximation to the Hessian of the merit function.

Both PGA and NPSOL are gradient based algorithms. They require derivatives of the objective function and of the constraints with respect to the control variables. The former vector of derivatives is called the **objective gradient** and the latter matrix of derivatives is called the **Jacobian**, or sensitivity, matrix. In addition, NPSOL constructs a second derivative matrix or **Hessian**.

One of the keys to optimization success is a well conditioned Jacobian matrix. Usually, this is formed by finite differencing. In IPOST, the finite difference control perturbations can be input by the user or computed by NPSOL. If NPSOL computes the perturbation size for each control parameter, the appropriate perturbation is that control value which produces the most accurate partial derivative for the control-constraint (or objective) combination, which is just above the numerical noise threshold. It can use either forward or central differencing techniques. Quite often about a third of the run is spent computing an accurate Jacobian because many simulation passes, or **function evaluations**, are needed to produce the finite difference perturbation. NPSOL also has the option of "verifying" an input perturbation size by constructing and comparing a finite differenced Jacobian.

# 12.0 TRAJECTORY DECOMPOSITION

A complete, end-to-end mission analysis for complex interplanetary missions with one or more intermediate flybys is not practical in one computational step. The complexities arising from gravity-assist and velocity change maneuvers present a formidable challenge to solve only one trajectory through the deterministic boundary value problem. Inclusion of all perturbations required for optimization and their subsequent cascade of multiple trajectory calculations would expand the problem to an unmanageable scope. Decomposition of the mission into tractable sub-problems has long been a viable alternative (Reference 12-1, 12-2).

## 12.1 EXPLICIT OPTIMIZATION - MASTER/SUBPROBLEMS

Decomposition is the resolution of the mission into sub-problems which are readily solved, followed by an iterative master level solution of approximate sub-problem solutions. The problem is described as a series of events and trajectory phases, specified by user input. The events can reflect physical happenings (starting time, time of periapsis, impulsive velocity change) or be informational events (time to record the state vector or a sphere-of-influence). Each trajectory phase starts and ends on an event. Complex missions are decomposed by partitioning the trajectory into subproblems of one or more phases. Sub-problems usually mirror a natural grouping of trajectory phases - such as a planet to planet leg.

Each subproblem is defined by a set of control (independent) parameters and a set of target (constraint or dependent) parameters. The controls are varied such that target conditions are met. This requires first generating a target sensitivity matrix, or Jacobian, ususally by finite differencing. In IPOST there are two ways to adjust the controls using the Jacobian matrix. One option is to employ a Newton-Raphson technique to "invert" the Jacobian. This is done with a L-U decomposition algorithm which does not require a full rank system, that is, the nimber of controls and constraints do not have to be equal. The second option employs NPSOL. NPSOL can be used to optimize a subproblem unique objective function at the same time meeting constraints. An example of subproblem controls would be midcourse $\Delta V$, and subproblem constraints would be planetary encounter conditions at a subsequent flyby. The master level problem supplies the starting independent variable values to each subproblem leg and receives in turn the targeted values.

The master problem, which represents the complete mission design, is the summation of the individual sub-problems. It controls optimization and specifies the individual sub-problem targets. The sub-problem solutions are relatively small in dimension. The master problem iterates through the sub-problem trajectory legs to optimize an overall target strategy in conjunction with the NPSOL optimization program. NPSOL evaluates the total trajectory in terms of target error and modifies the independent variables for the next master level pass.

## 12.2 IMPLICIT OPTIMIZATION - COLLOCATION

A more subtle form of decomposition is applied in the technique of collocation (Ref. 12-3, 12-4). This reformulation of the optimization problem requires setting up the trajectory as an implicit simulation where the trajectory is divided into many "independent" segments.

Implicit simulation computes the vehicle state between two junction points, or nodes. This trajectory segment can be a portion of a simulation phase or the entire phase. At nodes between events, the vehicle state must be the same for each side, that is, just prior to and just after the node. At nodes that correspond to events, the states on each side may be different. This discontinuity would occur if instantaneous event activities were specified, such as mass jettison or an impulsive $\Delta V$ maneuver.

The vehicle state is represented between nodes as Hermite, or cubic, polynomials. Each state component corresponds to a single Hermite polynomial. To compute the polynomial coefficients requires position, velocity, and acceleration (from evaluating the equations of motion) at each node. Figure 12 - 1 describes the methodology and compares it with explicit propagation.

In collocation, the states on each side of each node, that is, the pre- and post-node states, are allowed to vary. These free states become additional control variables.

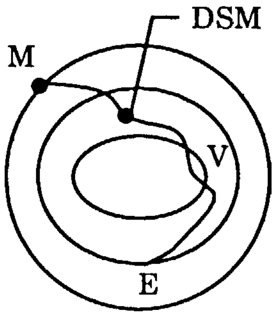
Additional constraints must also be introduced because the states have other imposed conditions. For example, the state on each side of an event node is related by the event activity. Thus, an additional constraint becomes the node connectivity between pre and post states, particularly due to any state discontinuity arising from a mission activity, such as an impulsive ΔV.

Another set of constraints is introduced because the state on each side of a node which is internal to a simulation phase, or between events, must be the same. In some collocation formulations, the zero state differences for these internal nodes are introduced as constraints. However in IPOST, the state on each side of an internal node is explicitly set equal to the state on the other side. This has the effect of reducing the number of controls and constraints such that only a single state at each internal node become control variables.

One other IPOST refinement eliminates the state differences of the first event node by explicitly setting the post-event state equal to the pre-event state plus any state discontinuity associated with activities in the first event. In theory, all double states can be reduced to single states by this procedure, but this remains a future modification.

There are other constraints introduced by collocation. These constraints involve forcing the Hermite polynomials to meet the equations of motion at the mid-point

Maximize $(\text{mass})_M$



| Master Problem Only | | Master + Subproblems |
|---|---|---|
| $T_E, \Delta\overline{V}_E, T_V, T_{DSM}, \Delta\overline{V}_{DSM}, T_z$<br>$(BDT, BDR, VINF)_V, (R_p, R_a, i)_M$ | Master Problem<br>• Controls<br>• Constraints | $T_E, T_V, T_{DSM}, T_M, (BDT, BDR, VINF)_V$<br>$(BDT, BDR, VINF)_V, (R_p, R_a, i)_M$ |
| N/A | Subproblem 1<br>• Controls<br>• Constraints | N/A |
| N/A | Subproblem 2<br>• Controls<br>• Constraints | N/A |

Figure 12-1. Explicit Optimization Decomposition Example

($t_c$ ) of each segment. The constraints, called defects, ($\vec{d}$) are formulated as:

$$\vec{d} = \dot{\vec{X}}\,(\vec{X}, tc) - \dot{\vec{X}}_{HP} = 0$$

$x \equiv [\dot{x}\,,\ \dot{y}\,,\ \dot{z}\,, \ddot{x}\,,\ \ddot{y}\,,\ \ddot{z}\,,\ \dot{m}\ ]$, the seven element vehicle state derivatives

where $\dot{\vec{X}}\ (\vec{X}, tc\ )$ = equations of motion evaluated at $t_c$ using Hermite polynomials (x) evaluated at

$\dot{\vec{X}}_{HP}$ = analytic derivations of Hermite polynomials, e.g.,

if $y = b_0 + b_1 t + b_2 t^2 + b_3 t^3$

then $\dot{y}_{HP} = b_1 + 2b_2 + 3b_3 t^2$

The net result of collocation is that for any reasonable simulation/optimization problem there are hundreds more control and constraint parameters than in an explicit simulation/optimization(see Figure 12 - 2). In the IPOST formulation, an equal number of controls and constraints are added via collocation. For example, if a classical problem has 10 controls and 6 constraints, and the simulation has 5 phases with 4 segments per phase, then the total number of controls and constraints are :

Number of controls = 10 + nc * 7 = 192
Number of constraints = 6 + nc * 7 = 188
where nc = (number of segments per phase + 1) (number of phases) + 1

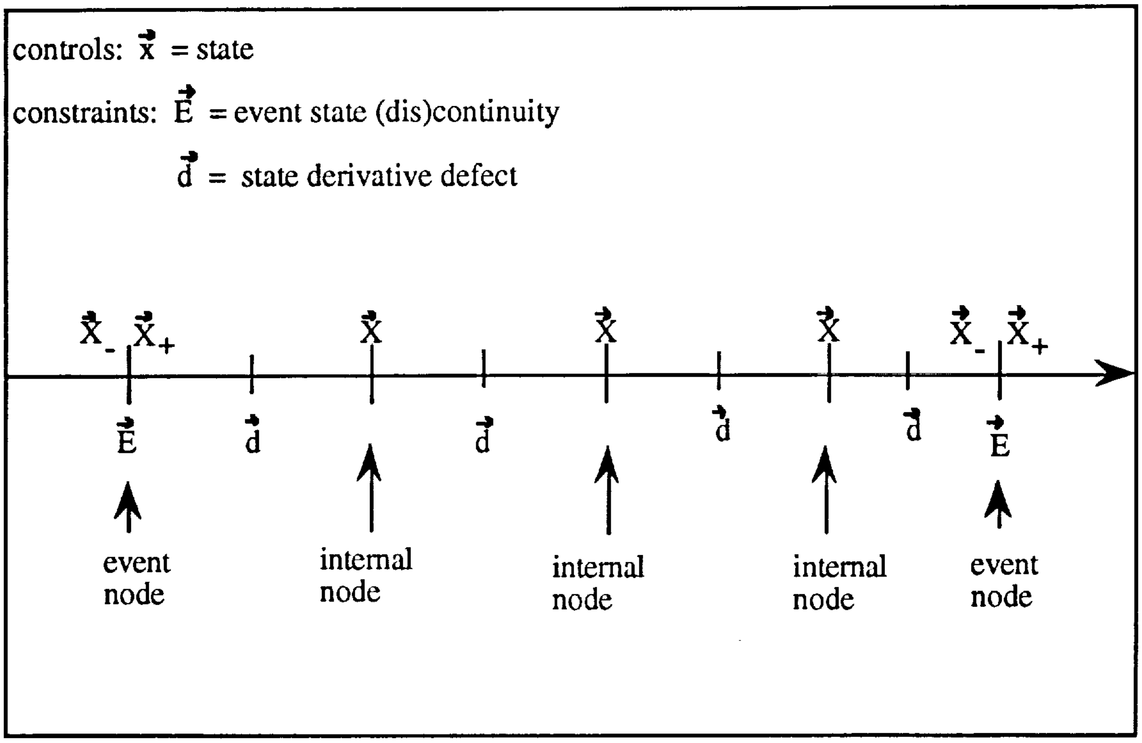


Figure 12 - 2. Controls and Constraints Introduced by Collocation

Although the dimensionality of the optimization problem has been greatly increased, there are several compensating features to collocation which ultimately expand its region of convergence. The first benefit is that each segment is relatively independent of the others because of the free states at each node. Thus, a change early in the mission is not amplified into extremely large, and unpredictable, changes later in the mission, as in explicit simulation/optimization.

Another collocation benefit is that most derivatives are analytic, and therefore rapidly, and accurately, computed. The few derivatives that must be done by finite differencing are of the "local" variety. That is, there is very little time mapping, at most over a single segment, which makes the finite differenced partials relatively stable and well-behaved.

Finally, the total Jacobian, although quite large, is a sparsely populated, banded, matrix. There are many techniques for efficient generation and manipulation of sparse matrices. In IPOST, the zero elements of the Jacobian are identified and pre-set, thus eliminating the wasteful application of finite differencing by NPSOL.

# 13.0 INTERPLANETARY TARGETING AND OPTIMIZATION OPTION (ITOO)

The optimization objective is defined as a function, $F(\vec{u})$, which is to be minimized over some time interval, $t_o$ to $t_f$, where $\vec{u}$ is an m x 1 set of control parameters (independent variables).

The targets are defined as a set of n x 1 constraint parameters (dependent variables)

$$\vec{y}(\vec{u}) = \vec{y}(\vec{u}, \vec{x}(\vec{u}), \vec{c}), \qquad [13\text{-}1]$$

where $\vec{x}(\vec{u}) = \vec{x}(\vec{u}), \vec{x}(t_0)$ are state variables (position, velocity, and perhaps mass) and c is the array of constraint values on $\vec{y}$.

The general targeting and optimization technique is a recursive application of $\vec{u}_{new} = \vec{u}_{old} + \Delta\vec{u}$, where $\Delta\vec{u}$ depends on $\Delta\vec{y}$ (a measure of how far away a target is from its desired value or constraint boundary) and the gradients $\partial F/\partial\vec{u}$ and $\partial\vec{y}/\partial\vec{u}$. The $\Delta u$ solution may also require past information on $\Delta\vec{u}$, $\Delta\vec{y}$, and/or the gradient matrices, plus numerous user-supplied weighting factors.

The problem solution starts with an initial guess of $\vec{u}_0$ and $\vec{x}'_0 = \vec{x}'(t_0)$. The equations of motion are integrated from $t_o$ to $t_f$ (or some user defined stopping condition that defines $t_f$). The $\Delta\vec{y}$ and F functions are evaluated. The cost and constraint gradients are computed. A control correction, $\Delta\vec{u}$, is computed and the new u is formed. The new conditions, $\vec{u}$ and $\vec{x}_0$, are propagated to form a new trajectory. This process is repeated until F is minimized (i.e., until $\partial F/\partial\vec{u} = 0$).

Assume that

$$F = \sum_{i=1}^{L}, w_i y_i^2, \qquad [13\text{-}2]$$

where L is the number of variables of interest, $w_i$ is a weighting factor, and $y_i$ is an unconstrained dependent variable and

$$\vec{y} = \vec{y}(\vec{u}, \vec{x}). \qquad [13\text{-}3]$$

From a truncated Taylor series expansion,

$$\Delta\vec{u} \approx -\left(\frac{\partial^2 F}{\partial u^2}\right)^{-1}\frac{\partial F}{\partial \vec{u}} \approx -\left(\sum_{i=1}^{L} w_i \left(\frac{\partial y_i}{\partial \vec{u}}\right)^T\left(\frac{\partial y_i}{\partial \vec{u}}\right)\right)^{-1}\left(\sum_{i=1}^{L} w_i y_i\left(\frac{\partial y_i}{\partial \vec{u}}\right)\right) \qquad [13\text{-}4]$$

This provides a direction of $\Delta\vec{u}$ and an initial estimate of its magnitude. This technique has special applications that will be shown later. It presumes that $\vec{y}$ is "small" and that $\vec{y}$ and $\vec{u}$ are linearly related. These conditions are quite often true in interplanetary situations.

The local Jacobian at time t is defined as $\partial\vec{z}(t)/\partial\vec{x}(t)$, with $\vec{z}$ representing various trajectory parameters, such as B-plane parameters, and $\vec{x}$ the state parameters (position and velocity). $\vec{z}$ can be control or target variables. The local Jacobian is computed by numerical differencing. Although explicit analytical partials exist for many interplanetary parameters, future growth dictates the need for the more generalized numerical differencing method.

There is a broad class of interplanetary problems which lends itself to be solved using the methods of ITOO. Figure 13-1 illustrates the situation. The initial and final times are specified. There are N bodies that are flown past. An impulsive trajectory correction maneuver is performed between each body. The final body often contains terminal conditions, such as planetary injection or small body rendezvous. Intermediate planets usually have constrained flyby conditions.

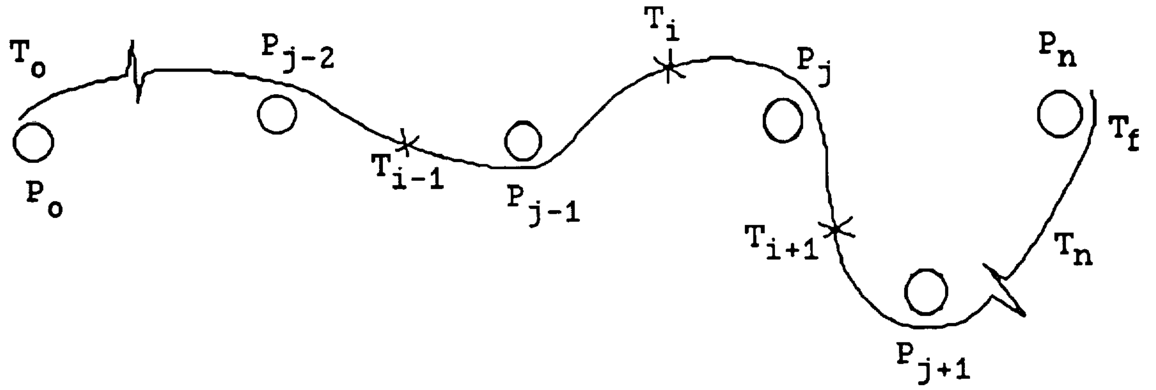


Figure 13-1. General Interplanetary Mission

Note that this geometry has applications beyond the transplanetary phases and includes satellite tours, such as the Galilean satellite tours, "cycler" missions that visit the same two bodies, such as Earth-Mars cyclers, and small body tours, such as multiple asteroid missions.

The targeting/optimization problem decomposes into two stages. The first stage is a simple targeting problem. The impulsive $\Delta\vec{V}$ of each maneuver is computed such that the following body's encounter conditions are met. There are three components of $\Delta\vec{V}$ and three target variables (user selected).

The second stage contains the optimization. The function to be optimized is identical to that discussed earlier. Here the dependent variables, $\vec{y}$, are the maneuver $\Delta\vec{V}$s. The independent variables, or control parameters, $\vec{u}$, are the maneuver times, $T_i$ , and the planetary encounter conditions, $\vec{P_j}$, which are given as

$$\vec{u} = \begin{pmatrix} \vec{P}_0 \\ T_1 \\ \vec{P}_1 \\ \vdots \\ T_i \\ \vec{P}_j \\ \vdots \\ T_n \\ \vec{P}_n \end{pmatrix} \qquad \vec{y} = \begin{pmatrix} \Delta\vec{V}_1 \\ \Delta\vec{V}_2 \\ \vdots \\ \Delta\vec{V}_1 \\ \vdots \\ \Delta\vec{V}_n \\ \Delta\vec{V}_{n+1} \end{pmatrix} , \qquad [13\text{-}5$$

and

$$F(\vec{y},\vec{u})) = \sum_{i=1}^{n+1} w_i \mid \vec{y}_i \mid^2 = \sum_{i=1}^{n+1} w_i \, \Delta\vec{V}_i^{\,T} \Delta\vec{V}_i \, . \qquad [13\text{-}6]$$

Here we note special requirements affecting the initial and final conditions. In transplanetary applications, there is typically a launch planet, $P_0$ , whose launch/escape conditions can constitute an additional set of control parameters. Also, there can be one or more terminal maneuvers at the terminal body, such as rendezvous or orbit capture, which must be included in the total cost function.

A reasonable upper bound for the number of bodies appears to be 15. This is consistent with all missions currently being projected for this century. The number of control variables is 4n+3 and the number of dependent variables is 3n+3. Hence, the maximum number of control and dependent parameters should be 63 and 48, respectively. In reality, IPOST limits control and dependent parameters to 45 each.

There are no constraints on the $\Delta\vec{V}$s because they are part of the cost function to be minimized. A few words on cost function weighting are in order.

If all the $w_i$s are unity, the cost function is simply the sum of squares of the $\Delta\vec{V}$ magnitudes. A more useful cost function might be the sum of the total $\Delta\vec{V}$

magnitudes. This can be done by setting $w_i = 1/ | \Delta\vec{V}_i |$. An even more useful function would be the sum of the total propellant expended (which is equivalent to maximizing delivered mass). This can be done by using the rocket equation,

$$wi = \frac{mi}{| \Delta\vec{V} |^2} \left(1\text{-}e^{-|\Delta\vec{V}|/Ci}\right) \qquad [13\text{-}7]$$

where $m_i$ is the S/C mass prior to $\Delta V$, and $C_i$ is the exhaust velocity and is given by

$$C_i = g_0 I_{sp_i}. \qquad [13\text{-}8]$$

There are, or can be, constraints imposed on the control variables. Typically, $\vec{P}_0$ is constrained by launch vehicle conditions, such as $C_3$ or declination. Maneuver times are usually restricted from occurring near a planet so as not to interfere with science data collection. Planetary encounter conditions are also constrained, such as not allowing closest approach beneath the planet's surface (impact).

Thus, we have a cost function dependent solely on unconstrained dependent variables, subject to constrained independent/control variables.

Whatever the optimization method, a necessary part is the gradients, $\partial F/(\partial \vec{u})$ and $\partial \vec{y}/\partial \vec{u}$ Because of the cost function formulation, the two gradients are related as follows

$$\frac{\partial F}{\partial \vec{u}} = 2 \sum_{i=1}^{n+1} w_i \, \Delta\vec{V}_i^{\,T} \frac{\partial \Delta\vec{V}_i}{\partial \vec{u}} \quad . \qquad [13\text{-}9]$$

We need only to compute $\partial\Delta\vec{V}_i/\partial \vec{u}$ and we have both cost and target gradients. The computations are even more efficient if we make use of analytic partials. Onestep, Multiconic and Encke propagators can produce analytic state transition matrices as part of their trajectory propagation process.

The state transition matrix is composed of four partitions,

$$\Phi_{i,i-1} \equiv \frac{\partial \vec{x}_i}{\partial \vec{x}_{i-1}} \equiv \begin{bmatrix} \dfrac{\partial \vec{R}_i}{\partial \vec{R}_{i-1}} & \dfrac{\partial \vec{R}_i}{\partial \vec{V}_{i-1}} \\ \dfrac{\partial \vec{V}_i}{\partial \vec{R}_{i-1}} & \dfrac{\partial \vec{V}_i}{\partial \vec{V}_{i-1}} \end{bmatrix} \qquad [13\text{-}10]$$

$\Phi_{j-1,i-1}$ , $\Phi_{i,j-1}$, $\Phi_{j,i}$,... are all output from the Onestep and Multiconic propagation process. We also note that

$$\Phi_{i,\, i-1} = \Phi_{i,\, j-1}\, \Phi_{j-1,\, i-1}. \qquad [13\text{-}11]$$

The local Jacobian can also be easily computed, and made available as

$$\frac{\partial \vec{P}_j}{\partial \vec{x}_j} \equiv \left[ \frac{\partial \vec{P}_j}{\partial \vec{R}_j} \middle| \frac{\partial \vec{P}_j}{\partial \vec{V}_j} \right] \qquad [13\text{-}12]$$

Subsequent chaining can generate other needed partials by

$$\frac{\partial \vec{P}_j}{\partial \vec{x}_j} = \frac{\partial \vec{P}_j}{\partial \vec{x}_j} \Phi_{i,\, j} \equiv \left[ \frac{\partial \vec{P}_j}{\partial \vec{R}_j} \middle| \frac{\partial \vec{P}_j}{\partial \vec{V}_j} \right] . \qquad [13\text{-}13]$$

The first step of the targeting/optimization problem is targeting. A simple iterative scheme can be employed where

$$\text{new } \Delta \vec{V}_i = \text{old } \Delta \vec{V}_i + \frac{\partial \vec{V}_i}{\partial \vec{P}_j} \Delta \vec{P}_j \quad , \qquad [13\text{-}14]$$

where

$$\frac{\partial \vec{V}_i}{\partial \vec{P}_j} = \left[ \frac{\partial \vec{P}_j}{\partial \vec{V}_i} \right]^{-1} , \qquad [13\text{-}15]$$

and $\Delta \vec{P}_j$ is the error in the encounter conditions (desired minus actual) from the last iteration. The partials, $\partial \vec{V}_j / \partial \vec{P}_j$, should be updated for each iteration.

Once targeting of each leg has been completed, the second stage is minimization of the cost function. Again, the chain rule can be applied. The results of $\partial \Delta \vec{V} / \partial \vec{u}$ are summarized below.

Define

$$S_{VRO}(i) \equiv \left.\frac{\partial \vec{V}_i}{\partial \vec{R}_i}\right|_{T_i, \vec{P}_i} = \left[\frac{\partial \vec{P}_j}{\partial \vec{V}_i}\right]^{-1} \frac{\partial \vec{P}_i}{\partial \vec{R}_i} , \qquad [13\text{-}16]$$

$$S_{VPO}(i) \equiv \left.\frac{\partial \vec{V}_i}{\partial \vec{P}_i}\right|_{T_i, \vec{R}_i} = \left[\frac{\partial \vec{P}_j}{\partial \vec{V}_i}\right]^{-1} , \qquad [13\text{-}17]$$

$$S_{RR1}(i) \equiv \frac{\partial \vec{R}_i}{\partial \vec{R}_{i-1}} + \frac{\partial \vec{R}_i}{\partial \vec{V}_{i-1}} S_{VRO}(i) , \qquad [13\text{-}18]$$

$$S_{VR1}(i) \equiv \frac{\partial \vec{V}_i}{\partial \vec{R}_{i-1}} + \frac{\partial \vec{V}_i}{\partial \vec{V}_{i-1}} S_{VRO}(i) , \qquad [13\text{-}19]$$

and

$$S_{VR2}(i) \equiv S_{VRO}(i)\, S_{RR1}(i) - S_{VR1}(i). \qquad [13\text{-}20]$$

For the sensitivities of $\Delta\vec{V}_i$ to maneuver times, $T_k$, where $i=1,\ldots,n+1$ and $k=1,\ldots,n$,

1) for $k > i$,

$$\frac{\partial\Delta\vec{V}_i}{\partial T_k} = 0 \qquad [13\text{-}21]$$

which means that all planets beyond the $i^{th}$ target planet have no effect on $\partial\Delta\vec{V}_i$ ;

2) for $k = i$,

$$\frac{\partial\Delta\vec{V}_i}{\partial T_i} = -S_{VRO}(i)\,\Delta\vec{V}_i\ ; \qquad [13\text{-}22]$$

3) for $k = i\text{-}1$,

$$\frac{\partial\Delta\vec{V}_i}{\partial T_{i-1}} = -S_{VR2}(i)\,\Delta\vec{V}_{i-1}\ ; \qquad [13\text{-}23]$$

4) for $k$ £ $i\text{-}2$,

$$\frac{\partial\Delta\vec{V}_i}{\partial T_k} = -S_{VR2}(i)\left[\prod_{L=k+1}^{i-1} S_{RR1}(L)\right]\Delta\vec{V}_k\ . \qquad [13\text{-}24]$$

For illustration, the full expression for $k = i\text{-}2$ is written

$$\frac{\partial\Delta\vec{V}_i}{\partial T_{i-2}} = -\left(\frac{\partial\vec{V}_i}{\partial\vec{R}_i}\left[\frac{\partial\vec{R}_i}{\partial\vec{R}_{i-1}} + \frac{\partial\vec{R}_i}{\partial\vec{V}_{i-1}}\frac{\partial\vec{V}_{i-1}}{\partial\vec{R}_{i-1}}\right] \cdot \right.$$

$$\left.\left[\frac{\partial\vec{V}_i}{\partial\vec{R}_{i-1}} + \frac{\partial\vec{V}_i}{\partial\vec{V}_{i-1}}\frac{\partial\vec{V}_{i-1}}{\partial\vec{R}_{i-1}}\right]\right)\left[\frac{\partial\vec{R}_{i-1}}{\partial\vec{R}_{i-2}} + \frac{\partial\vec{R}_{i-1}}{\partial\vec{V}_{i-2}}\frac{\partial\vec{V}_{i-2}}{\partial\vec{R}_{i-2}}\right]\Delta\vec{V}_{i-2}. \qquad [13\text{-}25]$$

Similarly, we can compute the sensitivities of $\Delta \vec{V}_i$ to planetary encounter conditions $\vec{P}_k$, where i=1,...,n+1 and k=0,1,...,n,

1) for k > i,

$$\frac{\partial \Delta \vec{V}_i}{\partial \vec{P}_k} = 0 \,; \qquad [13\text{-}26]$$

2) for k = i,

$$\frac{\partial \Delta \vec{V}_i}{\partial \vec{P}_i} = S_{VP0}(i) \,; \qquad [13\text{-}27]$$

3) for k = i-1,

$$\frac{\partial \Delta \vec{V}_i}{\partial \vec{P}_{i-1}} = \left[ S_{VRO}(i) \frac{\partial \vec{R}_i}{\partial \vec{V}_{i-1}} - \frac{\partial \vec{V}_i}{\partial \vec{V}_{i-1}} \right] S_{VPO}(i-1) \;; \qquad [13\text{-}28]$$

4) for k = i-2,

$$\frac{\partial \Delta \vec{V}_i}{\partial \vec{P}_{i-2}} = S_{VR2}(i) \frac{\partial \vec{R}_{i-1}}{\partial \vec{V}_{i-2}} \; S_{VPO}(i-2) \qquad [13\text{-}29]$$

5) for $k \leq$ i-3,

$$\frac{\partial \Delta \vec{V}_i}{\partial \vec{P}_k} = S_{VR2}(i) \left[ \prod_{L=k+1}^{i-1} S_{RR1}(L) \right] \frac{\partial \vec{R}_{k+1}}{\partial \vec{V}_k} \; S_{VPO}(k) \qquad [13\text{-}30]$$

Again, for illustration, the full expression for k = i-3 is written

$$\frac{\partial \Delta \vec{V}_i}{\partial \vec{P}_{i-3}} = \left( \frac{\partial \vec{V}_i}{\partial \vec{R}_i} \left( \frac{\partial \vec{R}_i}{\partial \vec{R}_{i-1}} + \frac{\partial \vec{R}_i}{\partial \vec{V}_{i-1}} \frac{\partial \vec{V}_{i-1}}{\partial \vec{R}_{i-1}} \right) - \left( \frac{\partial \vec{V}_i}{\partial \vec{R}_{i-1}} + \frac{\partial \vec{V}_i}{\partial \vec{V}_{i-1}} \frac{\partial \vec{V}_{i-1}}{\partial \vec{R}_{i-1}} \right) \right) \left( \frac{\partial \vec{R}_{i-1}}{\partial \vec{R}_{i-2}} + \frac{\partial \vec{R}_{i-1}}{\partial \vec{V}_{i-2}} \frac{\partial \vec{V}_{i-2}}{\partial \vec{R}_{i-2}} \right) \frac{\partial \vec{R}_{i-2}}{\partial \vec{V}_{i-3}} \frac{\partial \vec{V}_{i-3}}{\partial \vec{P}_{i-3}} \qquad [13\text{-}31]$$

We have now computed all of the partials needed to construct $\partial F / \partial \vec{u}$ and $\partial \vec{y} / \partial \vec{u}$ .

Optimization may cause one or more of the planetary legs to become untargeted. Thus, the targeting and optimization stages may have to be repeated until satisfactory convergence and optimization is achieved. The methodology of two-stage interplanetary targeting and optimization is consistent with the IPOST trajectory decomposition capability.

# 14.0 SPECIAL ONESTEP TARGETING

TARG1S is a restricted application of 1STEP which is a special adaptation of a generalized Multiconic method. This specialized targeting scheme is suitable for a restricted class of interplanetary problems. (See references 4-1 and 4-2 for restrictions.) An impulsive change in velocity ( $\Delta\vec{V}$ ) is computed at a specified maneuver time ($T_O$) to achieve target body relative conditions (targeting) at periapsis. Target conditions which work well are radius of closest approach (RCA), B-plane orbit angle ($\theta$), and time of closest approach ($T_P$). It basically solves a two point boundary value problem (TPBV) between the pseudostate position and desired target conditions (see Figure 14-1). In theory, this is more robust than solving the TPBV between initial velocity and target conditions because the nonlinear effects of the primary body propagation from initial state to pseudostate are minimized.

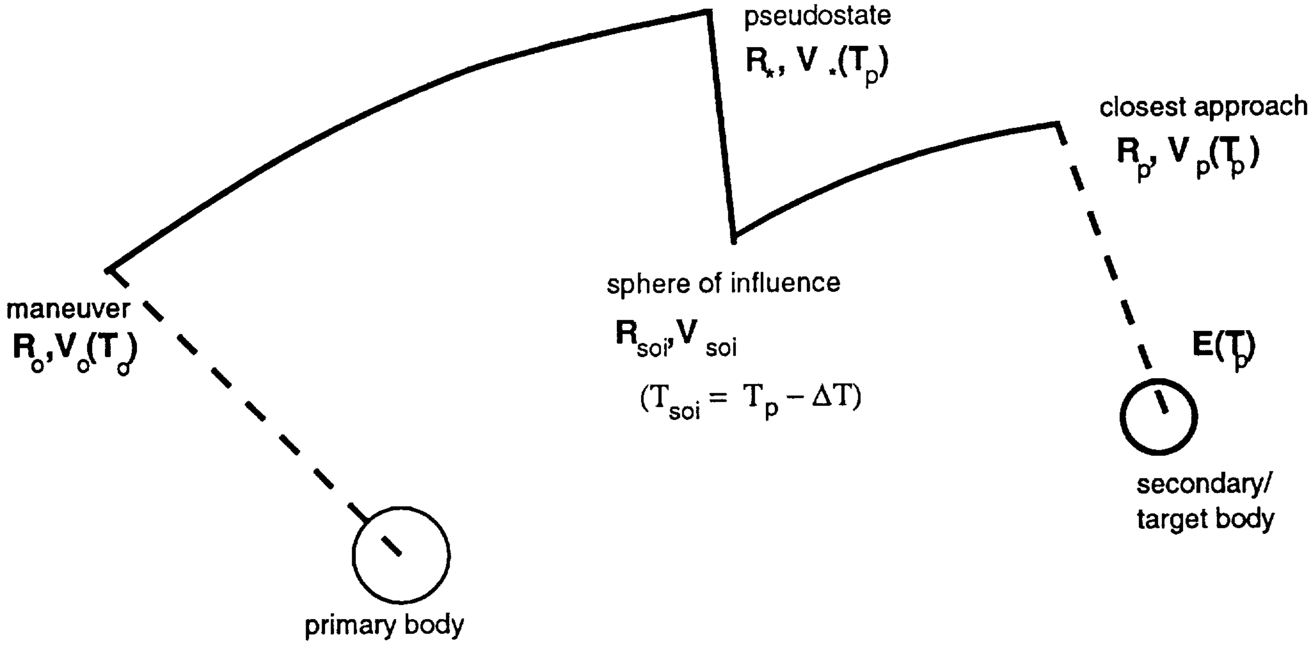


Figure 14-1. Special ONESTEP targeting

Define E as the target conditions at the desired time $T_p$. The target controls currently implemented are listed in the User's Manual. A change in the pseudostate position, R*, will affect E in two ways: directly and indirectly (through V*).

$$\delta E = \Bigg( \underbrace{\frac{\partial \vec{E}}{\partial \vec{X}_p} \frac{\partial \vec{X}_p}{\partial \vec{X}_{soi}} \frac{\partial \vec{X}_{soi}}{\partial \vec{X}_*} \frac{\partial \vec{X}_*}{\partial \vec{R}_*}}_{\text{direct}} + \frac{\partial \vec{E}}{\partial \vec{X}_p} \underbrace{\frac{\partial \vec{X}_p}{\partial \vec{X}_{soi}} \frac{\partial \vec{X}_{soi}}{\partial \vec{X}_*} \frac{\partial \vec{X}_*}{\partial \vec{V}_*} \frac{\partial \vec{V}_*}{\partial \vec{V}_0} \frac{\partial \vec{V}_0}{\partial \vec{R}_*} \Bigg) \delta \vec{R}_*}_{\text{indirect}}$$

We can compute $J = \partial \vec{E} / \partial \vec{X}_p$ the Jacobian by finite differencing. Furthermore,

$\dfrac{\partial \vec{X}_p}{\partial \vec{X}_{soi}} = \Phi_{ps}$ is the state transition matrix from $T_{soi}$ to $T_p$ from GOODYR.

$\dfrac{\partial \vec{X}_{soi}}{\partial \vec{X}_*} = \Phi_{s*}$ is the constant velocity transition matrix $= \begin{bmatrix} I & -I\Delta T \\ O & I \end{bmatrix}$

$$\frac{\partial \vec{X}_*}{\partial \vec{R}_*} = \begin{bmatrix} I \\ O \end{bmatrix} \quad \text{and} \quad \frac{\partial \vec{X}_*}{\partial \vec{V}_*} = \begin{bmatrix} O \\ I \end{bmatrix}$$

$\dfrac{\partial \vec{V}_*}{\partial \vec{V}_0}$ = lower right 3X3 portion of $\Phi_{*0}$ from GOODYR.

$\dfrac{\partial \vec{V}_0}{\partial \vec{R}_*}$ = lower left 3X3 portion of $\Phi_{*0}^{-1}$ from GOODYR.

If we define $A = \dfrac{\partial \vec{V}_*}{\partial \vec{V}_0} \dfrac{\partial \vec{V}_0}{\partial \vec{R}_*}$, then $\dfrac{\partial \vec{E}}{\partial \vec{R}_*} = (J) \ (\Phi_{sp}) \begin{pmatrix} I & - A\Delta T \\ & A \end{pmatrix}$

The targeting procedure is as follows.

(1) Start with initial primary body centered state $\vec{X}_0^T = \begin{pmatrix} \vec{R}_0 & \vec{V}_0 \end{pmatrix}$

and desired secondary body centered end conditions $E_d$.

(2) Propagate $\vec{X}_0$ to $T_p$ with respect to the primary body to obtain $\vec{X}_*^T = \left(\vec{R}_* \quad \vec{V}_*\right)$ and $\Phi_{*_0}$, $\Phi_{*_0}^{-1}$.

(3) Change pseudostate to secondary body frame of reference, and propagate back to $T_{soi} = T_p - \Delta T$ where $\Delta T$ = secondary body sphere of influence, with constant velocity, then propagate to $T_p$ with respect to the

secondary body to obtain $\vec{X}_p$ and $\Phi_{ps}$.

(4) Evaluate the actual target conditions, $\vec{E}_a$, from $\vec{X}_p$.

(5) If the target error $\Delta E = E_d - E_a$ meets desired tolerances, then exit targeting (go to Step 11). Otherwise, continue targeting (go to Step 6).

(6) Compute the Jacobian, $J = \partial\vec{E}/\partial\vec{X}_p$.

(7) Form $\vec{E}/\partial\vec{R}_*$, as expressed above.

(8) Compute $S = \left(\partial\vec{E}/\partial\vec{R}_*\right)^{-1}$ .

(9) The new $\vec{R}_*$ = old $\vec{R}_* + S\,\Delta E$ .

(10) Solve Lambert's problem from $\vec{R}_0$ to $\vec{R}_*$ with flight time of $T_p - T_0$ .

This provides a new initial velocity $\vec{V}_0$. Go to Step 2 and continue the targeting process.

(11) The total impulsive $\Delta V$ is the last $V_0$ - original $V_0$ .

# 15.0 SPECIAL FUNCTIONS

## 15.1 LAUNCH MANEUVERS

The launch maneuver sub-event in IPOST applies a maneuver from a parking orbit to a hyperbolic escape orbit. Computed outputs from this special subroutine are the new S/C mass (after escape burn), the $\Delta\vec{V}$ applied to complete the required maneuver, the burn duration, and the state of the S/C at periapsis of the hyperbolic escape orbit.

$a = -\frac{\mu}{V_\infty^2}$ = semi-major axis of the hyperbolic orbit where $V_\infty$ is the magnitude of the outgoing asymptote vector of the hyperbolic orbit. $V_\infty$ is either input or is calculated from

$$V_\infty = \sqrt{C3}$$

$$\hat{V}_\infty = \begin{bmatrix} \cos\delta & \cos\alpha \\ \cos\delta & \sin\alpha \\ \sin\delta & \end{bmatrix}$$

where

$\delta$ = desired declination,
$\alpha$ = desired right ascension,
$C3$ = desired escape energy.

If $\vec{V}_{\infty}$ is input, then

$$a = -\frac{\mu}{V_{\infty}^{2}}$$

Periapsis radius $R_p = h_p + R_s$

where

$h_p$ = circular park orbit altitude,
$R_s$ = surface radius.

$$b = R_p \sqrt{1 - \frac{2a}{R_p}}$$

$$\phi = \tan^{-1}\left(\frac{b}{| a |}\right) = \text{turn angle}$$

$$\hat{T} = \frac{V_{\infty} \times \hat{z}}{| V_{\infty} \times \hat{z} |}$$

$$\hat{R} = \hat{T} \times \hat{V}_{\infty}$$

If i is input, then

$$\hat{B} = \cos\theta\,\hat{T} + \sin\theta\,\hat{R}$$

$$\hat{R}_p = -\cos\phi\,\hat{V}_\infty + \sin\phi\,\hat{B}$$

$$\hat{N} = \hat{B} \times \hat{V}_\infty$$

$$\hat{V}_p = \hat{N} \times \hat{R}_p$$

$$V_p = \frac{b\ V_\infty}{R_p}$$

Outputs:

$$\vec{X}_p = \begin{bmatrix} R_p \hat{R} \\ V_p \hat{V}_p \end{bmatrix} = \text{escape state at periapsis}$$

$$\Delta V = V_p - \sqrt{\frac{\mu}{R_p}}$$

NOTE:

If F/(m g) < .02, then a low thrust spiral out is assumed and

$$\Delta V = \sqrt{\frac{\mu}{R_p} + V2_{\infty}}$$

ENDNOTE.

$$m_F = m_0 e^{-(\Delta V/(g*Isp))}$$

$$\Delta t = \frac{m_0\ \Delta V}{F}$$

where

Isp = specific impulse

F = average thrust during maneuver

g = reference gravitational acceleration

## 15.2 IMPULSIVE MANEUVER

The impulse maneuver sub-event in IPOST applies a maneuver, such as a midcourse correction, to the S/C state. Computed outputs from this subroutine are the mass of the S/C after the maneuver, the burn duration, and the in and out of plane ΔV angles. These are calculated as follows:

$$\Delta V = | \Delta \vec{V} |$$

$$m_F = m_0 \, e^{-(\Delta V/(g*Isp))}$$

where

$m_F$ = final S/C mass after the maneuver,

$m_0$ = initial S/C mass before the maneuver,

$g$ = surface gravitational acceleration of the earth,

$Isp$ = specific impulse.

$$\Delta m = m_F - m_0$$

$$\Delta t = \frac{m_0 \, \Delta V}{F} \quad = \text{burn duration,}$$

where

$F$ = average thrust during maneuver.

$$\text{Orbit normal } \hat{N} = \frac{\vec{R} \times \vec{V}}{| \vec{R} \times \vec{V} |}$$

$$\hat{\Delta V} = \frac{\Delta \vec{V}}{| \Delta \vec{V} |}$$

$$\beta = \cos^{-1} (\hat{\Delta V} \bullet \hat{N}) \text{ gives } 0 \le \beta' \le \pi$$

$$\therefore \beta = \frac{\pi}{2} - \beta \text{ gives } -\frac{\pi}{2} \le \beta \le \frac{\pi}{2}$$

where

$\beta$ = out of plane ΔV angle

$$\hat{T} = \hat{N} \times (\Delta\hat{V} \times \hat{N})$$

$\alpha'$ = directed angle from $\hat{V}$ to $\hat{T}$ about $\hat{N}$ gives $0 \leq \alpha' \leq 2\pi$

$$\left( \begin{array}{l} \text{If } \alpha' \leq \pi \text{ then } \alpha = \alpha' \\ \text{If } \alpha' > \pi \text{ then } \alpha = \alpha' - 2\pi \end{array} \right) \quad \begin{array}{c} \text{gives } -\pi < \alpha \leq \pi \\ \text{where } \alpha = \text{in plane } \Delta V \text{ angle} \end{array}$$

## 15.3 ORBITAL INSERTION

The orbital insertion maneuver sub-event in IPOST takes a S/C on a hyperbolic flyby trajectory and performs a maneuver to insert it into an orbit about a planet. Calculated outputs are total ΔV required for the maneuver, final mass after the maneuver, total burn duration, state of the S/C at periapsis of the new orbit, and the time from the current state to the final orbit periapsis. These are calculated as follows:

$$R_{ph} = a(1 - e)$$

where

$a$ = semi-major axis for the hyperbolic orbit,
$e$ = eccentricity for the hyperbolic orbit,
$R_{ph}$ = radius of periapsis for the hyperbolic orbit.

$$R_p = h_p + R_s$$

where

$h_p$ = desired periapsis altitude,
$R_s$ = surface radius of the planet,
$R_p$ = radius of periapsis of the desired orbit.
$R_a = h_a + R_s$

where

$h_a$ = desired apoapsis altitude,
$R_a$ = radius of apoapsis of the desired orbit.

$$a_1 = \frac{R_{ph} + R_a}{2}$$

$$a_2 = \frac{R_a + R_p}{2}$$

where

$a_1$ = semi-major axis of intermediate orbit after first ΔV,
$a_2$ = semi-major axis of the desired orbit.

$$V_{ph} = \sqrt{2\mu\left[\frac{2a - R_{ph}}{a\,R_{ph}}\right]}$$ = velocity at hyperbolic periapsis before the first maneuver,

$$V_{a_1} = \sqrt{2\mu\left[\frac{2a_1 - R_{ph}}{a_1\,R_{ph}}\right]}$$ = velocity at hyperbolic periapsis before the first maneuver,

If $i_D$ = desired inclination ≤ 0, and $R_a < R_{ph}$, then

$$\Delta V_1 = \sqrt{V_{ph}^2 + V_{a_1}^2 - 2\, V_{ph}\, V_{a_1} \cos\,(i_D - i)}$$

where

i = inclination of the hyperbolic orbit.

Otherwise,

$$\Delta V_1 = V_{ph} - V_{a_1}$$

$$\Delta t_1 = \frac{m_0 \, \Delta V_1}{F}$$

$$m_1 = m_0 e^{-\left(\frac{\Delta V_1}{g^* Isp}\right)}$$

where

F = average thrust during the maneuver,
$m_1$ = mass after the first maneuver.

$$V_{p_1} = \sqrt{2\,\mu\left[\frac{2a_1 - R_a}{a_1 R_a}\right]}$$ = velocity at apoapsis of transfer before the second maneuver,

$$V_{p_2} = \sqrt{2\,\mu\left[\frac{2a_2 - R_a}{a_2 R_a}\right]}$$ = velocity at apoapsis of desired after the second maneuver.

If $i_D$ = desired inclination ≤ 0, and $R_a < R_{ph}$, then

$$\Delta V_2 = \sqrt{V_{p1}^2 + V_{a1}^2 - 2\,V_{p1}\,V_{a_1}\cos\left(i_D - i\right)}$$

where

i = inclination of the hyperbolic orbit.

Otherwise,

$$\Delta V_2 = V_{p_1} - V_{a_2}$$

$$\Delta t_2 = \frac{m_1 \, \Delta V_2}{F}$$

$$m_2 = m_1 e^{-\left(\frac{\Delta V_2}{g^* Isp}\right)}$$

where

$F$ = average thrust during the maneuver,
$m_1$ = mass after the first maneuver.

The transfer time, TF, from the hyperbolic periapsis to the desired orbit periapsis is given by

$$TF = \frac{\pi}{\sqrt{\mu}} \left[ \sqrt{a_1^3} + \sqrt{a_2^3} \right] \quad .$$

The total $\Delta V$ is therefore given as

$$\Delta V = \Delta V_1 + \Delta V_2 \quad .$$

NOTE: If the value $[ F / m_0 g ] < .02$ then the maneuver is a low thrust spiral in, and therefore

$$\Delta V = \sqrt{(\mu/R_p) - \mu/a}$$

$$TF = \Delta t = m_0 \Delta V / F$$

$$m_f = m_0 e^{-(\Delta V/g*Isp)}$$

$$a_2 = R_p$$

If mean anomaly, M, is greater than zero, then M = -M,

$$t_p = -M/n + TF$$

where

$$n = \sqrt{\mu \ /\text{-}a^3}.$$

Compute the final orbit state at periapsis, $\vec{x}_p$,

Convert orbital elements with M = 0, to position, $\vec{r}$, and velocity, $\vec{v}$.

$$\vec{R}_p = (R_p / R_{ph})\,\vec{r}$$

$$\vec{V}_p = (V_p / V_{ph})\,\vec{v}$$

where

$$V_p = \sqrt{2\mu\ (2a_2 - R_p)\ /\ a_2 R_p)}$$

Therefore

$$\vec{x}_p = \begin{pmatrix} \vec{R}_p \\ \vec{V}_p \end{pmatrix} .$$

## 15.4 SPACECRAFT MASS

The N-stage mass properties model from POST has been adapted for IPOST. The mass of the spacecraft ($m_{s/c}$) after an event (positive side) is calculated from the mass before the event (minus side) as

$$m^{+}_{s/c} = m^{-}_{s/c} - \left(m_{Jett} + \lambda_D\, m_{prop}\right) - \lambda_s\, \Delta m$$

where

$m_{Jett}$ = mass to be jettisoned,<br>
$\lambda_D$ = a factor to specify the fraction of propellant mass to be dumped,<br>
$m_{prop}$ = current (remaining) propellant mass<br>
$\lambda_s$ = propellant use (scale) factor<br>
$\Delta m$ = change is propellant mass due to propulsive accelerations (Section 8.2)

Finite thrust propulsion algorithms model low thrust, generalized thrust, and blowdown systems. These models return a mass flow rate which is integrated to obtain Dm using a simple Euler integration. Before each integration step the propellant ($m_{prop}$) is tested to determine if sufficient fuel exists for the next step.

Impulsive maneuvers (launch, impulse, and orbital insertion) directly change $m_{s/c}$ (Sections 15.1, 15.2, 15.3, respectively).

# 16.0 REFERENCES

1-1. "Interplanetary Program to Optimize Simulated Trajectories," Final Report, Volumes I, II, III, Fitzgerald, Hong, Kent, Milleur, and Olson, Martin Marietta Corporation, March 1990.

3-1. "A General Method for the Computation of Cartesian Coordinates and Partial Derivatives of the Two Body Problem," W.H. Goodyear, NASA CR-522, September 1966.

3-2. "A Unified Form of Lambert's Theorem," E.R.Lancaster and R.C. Blanchard, NASA TN D-5368, Sept. 1969.

4-1. "A Pseudostate Theory for the Approximation of Three-Body Trajectories," Wilson, S. W., AIAA Paper 70-1061, August 1970.

4-2. "Application of the Pseudostate Theory to the Three-body Lambert Problem," D.V. Byrnes, AAS 79-163, June 1979.

5-1. "Applications of an N-body Reference Orbit," K. Stumpff, E.H. Weiss, Journal of Astronautical Sciences, Sept-Oct 1968.

5-2. "Minimum Impulse Three-Body Trajectories," D'Amario, L. A., Ph.D. Dissertation, Massachusetts Institute of Technology, May, 1973.

5-3. "Updated User's Guide for MULCON," Kwok, J. H. and Nacozy, P. E., Jet Propulsion Laboratory Contract #955140 performed at The University of Texas at Austin, 1 January 1979.

6-1 "An Introduction to the Mathematics and Methods of Astrodynamics," Richard H. Battin, AIAA Education Series, 1987.

6-2. "MAPSEP (Mission Analysis Program for Solar Electric Propulsion), Volume III (Program Manual)", P.Hong, et al, Martin Marietta Corporation, April 1980.

9-1. "MAPSEP (Mission Analysis Program for Solar Electric Propulsion), Volume I (Analytic Manual), Volume II (User's Manual), Volume III (Program Manual)", P.Hong, et al, Martin Marietta Corporation, Feb. 1980.

10-1. "CPDS for Astrodynamics Common (CPCI 225), Appendix II," DSM Document CG-SCF-225A, 5 October 1983.

11-1. "User's Guide for NPSOL (Version 4.1): A Fortran Package for Nonlinear Programming", P.E. Gill, W. Murray, M.A. Saunders, M.H. Wright, Stanford University Department of Operations Research, Sept. 1984.

11-2. "User's Guide for NPSOL (Version 4.1): A Fortran Package for Nonlinear Programming", P.E. Gill, W. Murray, M.A. Saunders, M.H. Wright, Stanford University Department of Operations Research, Sept. 1984.

12-1. "A Two-Level Trajectory Decomposition Algorithm Featuring Optimal Intermediate Target Selection", F. M. Peterson, D. E. Cornick, G. L. Brauer, Journal of Spacecraft, Nov 1977.

12-2. "A New Method for Optimizing Multiple-Flyby Trajectories", L. A. D'Amario, D. V. Byrnes, R. H. Stanford, Journal of Guidance and Control, Nov-Dec 1981.

12-3. "Minimum-Fuel Aeroassisted Noncoplanar Orbital Transfer Using Collocation and Nonlinear Programming", Y.Y. Shi, D.H. Young, R.L. Nelson, AIAA, 1990.

12-4. "A collocation method for boundary value problems," R. Russell and L. Shampine, Numerical Math, # 19, 1972.

13-1. "A New Method for Optimizing Multiple-Flyby Trajectories", L. A. D'Amario, D. V. Byrnes, R. H. Stanford, Journal of Guidance and Control, Nov-Dec 1981.

13-2. "Interplanetary Trajectory Optimization with Applications to Galileo",L. A. D'Amario, D. V. Byrnes, R. H. Stanford, Journal of Guidance and Control, Sep-Oct 1982.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188), Washington, DC 20503.

**1. AGENCY USE ONLY** *(Leave blank)*

**2. REPORT DATE**
October 1992

**3. REPORT TYPE AND DATES COVERED**
Contractor Report

**4. TITLE AND SUBTITLE**
Interplanetary Program To Optimize Simulated
Trajectories (IPOST)
Volume II - Analytic Manual

**5. FUNDING NUMBERS**
C NAS1-18230
WU 506-49-11-02

**6. AUTHOR(S)**
P. E. Hong, P. D. Kent, D. W. Olson, and C. A. Vallado

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**
Martin Marietta Astronautics
Space Launch Systems Company
P. O. Box 17
Denver, CO 80201

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**
National Aeronautics and Space Administration
Langley Research Center
Hampton, VA 23681-0001

**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
NASA CR-189653,
Volume II (Revised)

**11. SUPPLEMENTARY NOTES** Supersedes NASA CR-189653, Volume II, July 1992.
Langley Technical Monitor: Richard W. Powell
Final Report

**12a. DISTRIBUTION/AVAILABILITY STATEMENT**
Unclassified - Unlimited

Subject Category 16

**12b. DISTRIBUTION CODE**

**13. ABSTRACT** *(Maximum 200 words)*

IPOST is intended to support many analysis phases, from early interplanetary feasibility studies through spacecraft development and operations. The IPOST output provides information for sizing and understanding mission impacts related to propulsion, guidance, communications, sensor/actuators, payload, and other dynamic and geometric environments. IPOST models three degree of freedom trajectory events, such as launch/ascent, orbital coast, propulsive maneuvering (impulsive and finite burn), gravity assist, and atmospheric entry. Trajectory propagation is performed using a choice of Cowell, Encke, Multiconic, Onestep, or Conic methods. The user identifies a desired sequence of trajectory events, and selects which parameters are independent (controls) and dependent (targets), as well as other constraints and the cost function. Targeting and optimization is performed using the Stanford NPSOL algorithm. IPOST structure allows sub-problems within a master optimization problem to aid in the general constrained parameter optimization solution. An alternate optimization method uses implicit simulation and collocation techniques.

**14. SUBJECT TERMS**
Collocation, targeting, trajectory propagation

**15. NUMBER OF PAGES**
67

**16. PRICE CODE**
A04

**17. SECURITY CLASSIFICATION OF REPORT**
Unclassified

**18. SECURITY CLASSIFICATION OF THIS PAGE**
Unclassified

**19. SECURITY CLASSIFICATION OF ABSTRACT**

**20. LIMITATION OF ABSTRACT**